PROJECTS ■ THEORY ■ APPLICATIONS ■ CIRCUITS ■ TECHNOLOGY

NUTS AND VOLTS

www.nutsvolts.com
February 2010

EVERYTHING FOR ELECTRONICS

TouchTone Phone Controller

Activate multiple AC devices by phone via wireless transmitters

A Quick Tour And More Experiments With The 16-Bit Experimenter Board

✦ **Open Communication**
Go wireless OR wired with your home network.

✦ **Digital Time Piece**
Versatile, Customizable
Pick and choose the features and style you want to customize your own digital alarm clock.

✦ **Experiments With The Whirlybird Wind Turbine**
• Three-Phase Rectification
• Wind Speed, RPM, and Power Output
• The Effects of Magnetic Coupling
• Star versus Delta Windings
• Hands Off Braking
• And More!

U.S. $6.50 CANADA $7.50

All displays shown are actual size

SerialLCD™

NetMedia

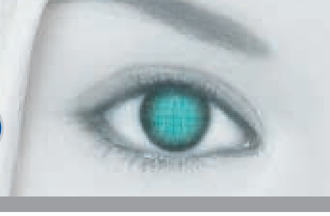
BasicX.com

RoHS 2002/95/EC COMPLIANT

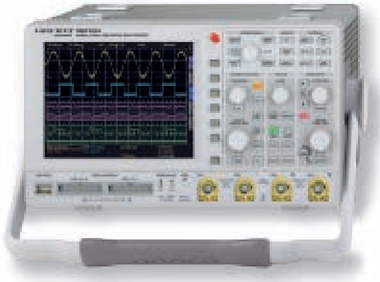
NetMedia Inc., 10940 N Stallard Pl, Tucson Arizona 85737 tel.520.544.4567 fax.520.544.0800 www.netmedia.com

GREAT VALUE IN TEST & MEASUREMENT

Call for educational discount



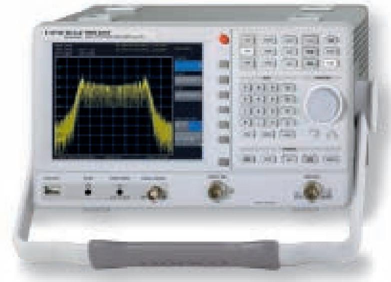
350 MHZ 2/4 CHANNEL DIGITAL OSCILLOSCOPE HMO3522/HMO3524



USB · USB Stick · optional IEEE-488 · optional TCP/IP · DVI Output · inclusive

- ☑ 4 GSa/s Real time, 50 GSa/s Random sampling, low noise flash A/D converter (reference class)
- ☑ 2 MPts memory per channel, memory Zoom up to 100,000:1
- ☑ MSO (Mixed Signal Opt. HO3508/HO3516) with 8/16 logic channels
- ☑ Vertical sensitivity 1 mV...5 V/div. (into 1 MΩ/50 Ω) Offset control ±0.2...±20 V
- ☑ 12 div. x-axis display range
  20 div. y-axis display range with **V**irtual**S**creen function
- ☑ Trigger modes: slope, video, pulsewidth, logic, delayed, event
- ☑ FFT for spectral analysis
- ☑ Lowest noise fan
- ☑ 6 digit counter, Autoset, automeasurement, formula editor, ratiocursor
- ☑ Crisp 6.5" TFT VGA display, LED backlight, DVI output

3 GHZ SPECTRUM ANALYZER HMS3000/HMS3010



1GHz: HMS1000 HMS1010

USB · USB Stick · optional IEEE-488 · optional TCP/IP · DVI Output · inclusive

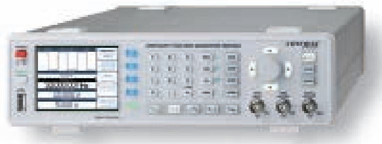
- ☑ Frequency range 100 kHz...3 GHz
- ☑ Amplitude measurement range -114...+20 dBm
  DANL -135dBm with Preamp. Option HO3011
- ☑ Sweep time 20 ms...1000 s
- ☑ Resolution bandwidth 100 Hz...1 MHz in 1–3 steps, 200 kHz (-3 dB)
  additional 200 Hz, 9 kHz, 120 kHz, 1 MHz (-6 dB)
- ☑ Spectral purity < -100 dBc/Hz (@100 kHz)
- ☑ Video bandwidth 10 Hz...1 MHz in 1–3 steps
- ☑ Tracking Generator (HMS3010) -20 dBm/0 dBm
- ☑ Integrated AM and FM demodulator (int. speaker)
- ☑ Detectors: Auto-, min-, max-peak, sample, RMS, quasi-peak

PROGR. 2/3/4 CHANNEL HIGH-PERFORMANCE POWER SUPPLY HMP SERIES

USB · RS-232 · optional IEEE-488 · optional TCP/IP · inclusive

- ☑ HMP2020: 1x0...32 V/0...10 A 1x0...5.5 V/0...5 A, max. 188 W
- ☑ HMP2030: 2x0...32 V/0...5 A 1x0...5.5 V/0...5 A, max. 188 W
- ☑ HMP4030: 3x0...32 V/0...10 A, max. 384 W
- ☑ HMP4040: 4x0...32 V/0...10 A, max. 384 W
- ☑ 188/384W output power realized by intelligent power management
- ☑ Low residual ripple: < 150 μV_{rms} due to linear post regulators
- ☑ High setting- and read-back resolution of up to 1 mV/0.2 mA
- ☑ HMP4030/HMP4040: Keypad for direct parameter entry
- ☑ Galvanically isolated, earth-free and short circuit protected output channels
- ☑ Advanced parallel- and serial operation via V/I tracking
- ☑ EasyArb function for free definable V/I characteristics
- ☑ FuseLink: individual channel combination of electronic fuses
- ☑ Free adjustable overvoltage protection (OVP) for all outputs
- ☑ All parameters clearly displayed via LCD/glowing buttons

25/50 MHZ ARBITRARY FUNCTION GENERATOR HMF2525/HMF2550



USB · USB Stick · optional IEEE-488 · optional TCP/IP · RS-232 · inclusive

- ☑ Frequency range 10 µHz...25 MHz/50 MHz
- ☑ Output voltage 5 mV_{pp}...10 V_{pp} (into 50 Ω) DC Offset ±5 mV...5 V
- ☑ Arbitrary waveform generator: 250 MSa/s, 14 Bit, 256 kPts
- ☑ Sine, Square, Pulse, Triangle, Ramp, Arbitrary
  waveforms incl. standard curves (white, pink noise etc.)
- ☑ Total harmonic distortion 0.04 % (f < 100 kHz)
- ☑ Burst, Sweep, Gating, external Trigger
- ☑ Rise time <8 ns, in pulse mode 8...500 ns variable-edge-time
- ☑ Pulse mode: Frequency range 100 µHz...12.5 MHz/25 MHz,
  pulse width 10 ns...999 s, resolution 5 ns
- ☑ Modulation modes AM, FM, PM, PWM, FSK (int. and ext.)
- ☑ 10 MHz Timebase: ±1ppm TCXO, rear I/O BNC connector
- ☑ Front USB connector: save & recall of set-ups and waveforms
- ☑ 3.5" TFT: crisp representation of the waveform and all parameters

LCR-BRIDGE HM8118

incl. HZ188

USB · RS-232 · optional IEEE-488

- ☑ Basic Accuracy 0.05 %
- ☑ Measurement functions L, C, R, |Z|, X, |Y|, G, B, D, Θ, Δ, D, M, N
- ☑ Test frequencies 20 Hz...200 kHz
- ☑ Up to 12 measurements per second
- ☑ Parallel and Series Mode
- ☑ Binning Interface HO118 (optional) for automatic sorting of components
- ☑ Internal programmable voltage and current bias
- ☑ Transformer parameter measurement
- ☑ External capacitor bias up to 40 V
- ☑ Kelvin cable and 4 wire SMD Test adapter included in delivery
- ☑ Galvanically isolated USB/RS-232 Interface, optional IEEE-488

1,2 GHZ/3 GHZ RF-SYNTHESIZER HM8134-3/HM8135

USB · RS-232 · optional IEEE-488

- ☑ Outstanding Frequency range 1 Hz...1,2 GHz/3 GHz
- ☑ Output power -127...+13 dBm/-135...+13 dBm
- ☑ Frequency resolution 1 Hz (accuracy 0.5 ppm)
- ☑ Input for external time base (10 MHz)
- ☑ Modulation modes: AM, FM, Pulse, Φ, FSK, PSK
- ☑ Rapid pulse modulation: typ. 200 ns
- ☑ Internal modulator (sine, square, triangle, sawtooth) 10 Hz...150 kHz/200 kHz
- ☑ High spectral purity
- ☑ Standard: TCXO (temperature stability: $\pm 0.5 \times 10^{-6}$)
  Optional: OCXO (temperature stability: $\pm 1 \times 10^{-8}$)
- ☑ Galvanically isolated USB/RS-232 Interface, optional IEEE-488
- ☑ 10 configuration memories including turn-on configuration

HAPRO Electronics
Tel: +1-516-794-4080 · Fax: +1-516-794-1855 · sales@haproelectronics.com

www.hameg.com

HAMEG Instruments
A Rohde & Schwarz Company

AUDIO KITS FOR ELECTRONIC ENTHUSIASTS

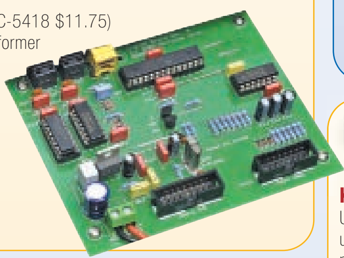
STEREO DIGITAL TO ANALOG CONVERTER KIT

KC-5487 $90.75 plus postage & packing

If you listen to CDs through a DVD player, you can get sound quality equal to the best high-end CD players with this DAC kit. It has one coaxial S/PDIF and two TOSLINK inputs to connect a DVD player, set-top box, DVR, computer or any other linear PCM digital audio source. It also has stereo RCA outlets for connection to a home theatre or hi-fi amplifier. This kit requires some SMD soldering skills. See website for full specifications.

- Short form kit with I/O, DAC and switch PCB and on-board components only.
- Requires: PSU (KC-5418 $11.75) and toroidal transformer

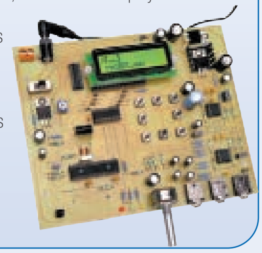
MORE INFO AVAILABLE ONLINE



SD CARD SPEECH RECORDER & PLAYER KIT

KC-5481 $43.50 plus postage & packing

Use this kit to store your WAV files on MMC/SD/SDHC cards. It can be used as a jukebox, a sound effects player or an expandable digital voice recorder. You can use it as a free-standing recorder or in conjunction with any Windows, Mac or Linux PC. Short form kit includes overlay PCB, SD card socket and electronic components.



AV SIGNAL BOOSTER KIT

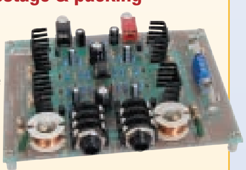
KC-5350 $63.50 plus postage & packing

You may experience some signal loss when using long AV cables. This kit will boost your composite, S-Video and stereo audio signals, preserving them for the highest quality transmission to your home theatre, projector or large screen TV. Kit includes case, PCB, silk-screened punched panels and all electronic components with clear English instructions. Requires 9VAC wall adaptor.

STEREO HEADPHONE DISTRIBUTION AMPLIFIER KIT

KC-5417 $20.50 plus postage & packing

Drives one or two stereo headphones from any line level (1 volt peak to peak) input. The circuit features a facility to drive headphones with impedances from about 8-600Ω. Comes with PCB and all electronic components.

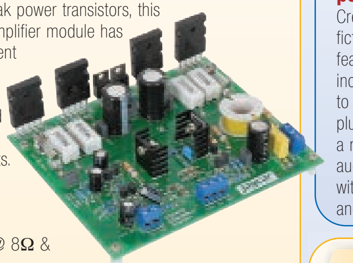


Also recommended: Box HB-6012 $3.50
Power Supply Kit KC-5418 $11.75

ULTRA-LOW DISTORTION AMPLIFIER MODULE KIT

KC-5470 $55.25 plus postage & packing

Using new ThermalTrak power transistors, this ultra-low distortion amplifier module has no need for a quiescent current adjustment or a Vbe multiplier transistor. Kit supplied with PCB and all electronic components. Heat sink and power supply not included.

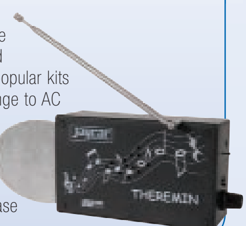


Output: 135WRMS @ 8Ω & 200WRMS @ 4Ω
Frequency response: 4Hz - 50kHz, 1m
Harmonic distortion: <0.008% @ 20Hz-20kHz

THEREMIN SYNTHESIZER KIT - MKII

KC-5475 $43.50 plus postage & packing

Create your own eerie science fiction sound effects! Updated features to one of our most popular kits include extra test points, change to AC to avoid switchmode plugpack interference, and a new skew control to vary audio tone. Contains PCB with overlay, pre-machined case and all specified components.



BRIDGE MODE ADAPTOR KIT FOR STEREO AMPLIFIERS

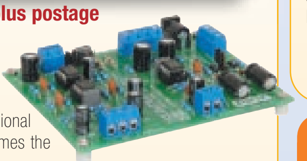
KC-5469 $16.25 plus postage & packing

Enables you to run a stereo amplifier in 'Bridged Mode' to effectively double the power available to drive a single speaker. There are no modifications required on the amplifier as this clever kit does the signal processing. Supplied with silk screened PCB and components. Requires balanced (+/-) power supply.

BALANCED TO UNBALANCED AUDIO CONVERTER KIT

KC-5468 $19.25 plus postage & packing

Standard audio gear does not have the balanced inputs and outputs found in professional systems. This kit overcomes the problem by adapting an unbalanced input to balanced output and vice versa. This allows domestic equipment to be integrated into a professional installation while maintaining the inherent high immunity to noise pick-up on long cable runs provided by balanced lines.



- PCB and all electronic components included
- ±9-15VDC 9-30VDC 7-12VAC

50 WATT AMPLIFIER MODULE KIT

KC-5150 $17.50 plus postage & packing

A single chip module that provides 50WRMS @ 8Ω with very low distortion. PC board and all electronic components supplied. PC board size only 84 x 58mm. Requires heatsink. See website for full specs.

Heatsink to suit HH-8590 $10.00

MORE INFO AVAILABLE ONLINE

FREE CATALOG

Checkout Jaycar's extensive range

We have kits & electronic projects for use in:

- Audio & Video
- Car & Automotive
- Computer
- Lighting
- Power
- Test & Meters
- Learning & Educational
- General Electronics Projects
- Gifts, Gadgets & Just for fun!

For your FREE catalog logon to www.jaycar.com/catalog or check out the range at www.jaycar.com

POST & PACKING CHARGES

Order Value	Cost
$25 - $49.99	$7.50
$50 - $99.99	$20
$100 - $199.99	$40
$200 - $499.99	$60
$500+	$75

Max weight 12lb (5kg). Heavier parcels POA. Minimum order $25.

Note: Products are dispatched from Australia, so local customs duty & taxes may apply. Prices valid until 28/2/10

HOW TO ORDER

- ORDER ON-LINE: www.jaycar.com
- PHONE: 1-800-784-0263
- FAX: +61 2 8832 3118*
- EMAIL: techstore@jaycar.com
- POST: P.O. Box 107, Rydalmere NSW 2116 Australia
- ALL PRICING IN US DOLLARS
- MINIMUM ORDER ONLY $25

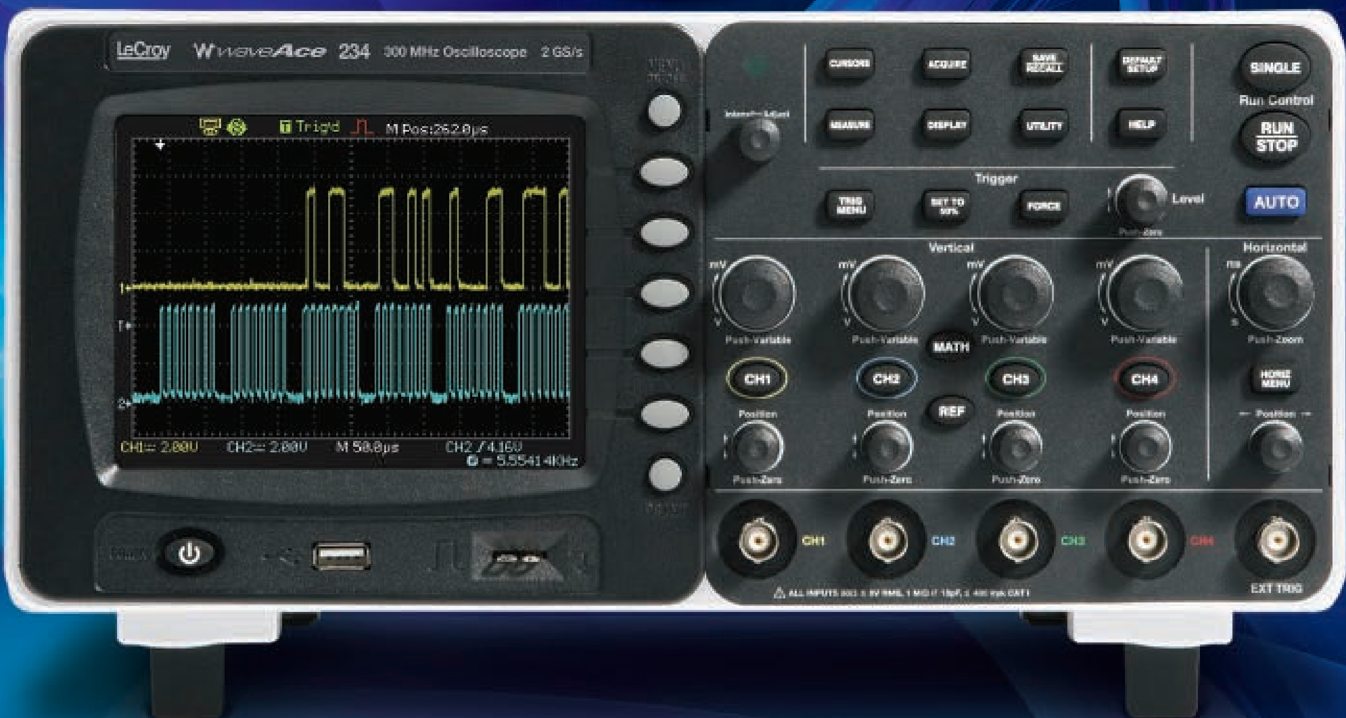
*Australian Eastern Standard Time (Monday - Friday 09.00 to 17.30 GMT + 10 hours only)
Expect 10-14 days for air parcel delivery

1-800-784-0263 www www.jaycar.com

jaycar Electronics

Debug With Confidence

Starting at
$695



LeCroy WaveAce™ Oscilloscopes
40 MHz - 300 MHz

- 40 MHz, 60 MHz, 100 MHz, 200 MHz and 300 MHz bandwidths
- 2 and 4 channel models available
- Sample rates up to 2 GS/s
- Longest memory in class - up to 10 kpts/Ch (20 kpts interleaved)
- 5.7" bright color display on all models
- 32 automatic measurements
- 4 math functions plus FFT
- Large internal waveform and setup storage
- Multi-language user interface and context sensitive help
- USB connections for memory sticks, printers and PCs

LeCroy Authorized Distributors:

Avalon Equipment Corp.
(888)-542-8256
avalontest.com

Trek Equipment
(888) 873-5374
trekequipment.com

Stanley Supply & Services
(800) 225-5370
StanleySupplyServices.com

Tequipment.NET
(877) 742-8378
tequipment.net

Kemp Instruments
(888) 619-2266
kempinstruments.com

Meunier Elect. Supply
(800) 638-6437
meunierusa.com

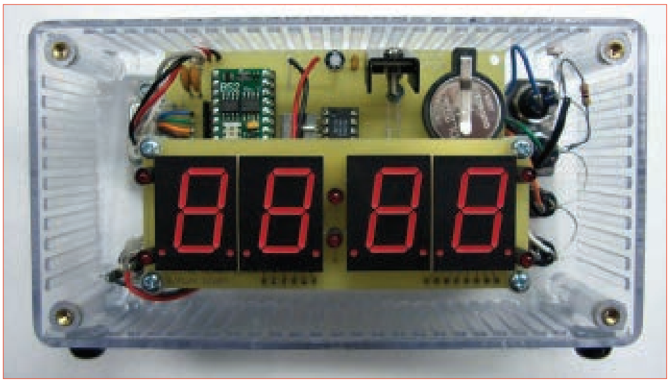
LeCroy

To learn more about LeCroy's full line of oscilloscopes, including the 30 GHz WaveMaster 8 Zi, the world's fastest real-time oscilloscope, visit www.lecroy.com or call 1-800-5-LeCroy

February

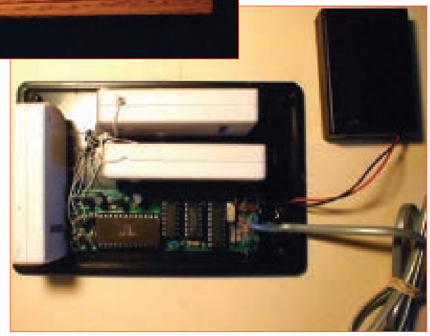
Nuts & Volts 2010

www.nutsvolts.com



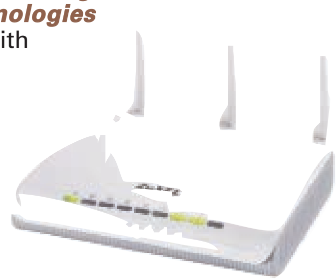
Page 26

Page 37

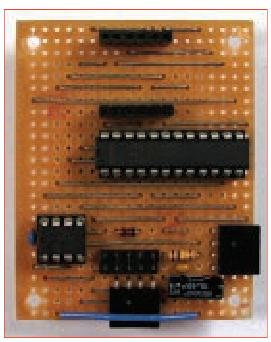


Projects & Features

Columns



Page 13



Page 56

Departments

Nuts & Volts (ISSN 1528-9885/CDN Pub Agree #40702530) is published monthly for $26.95 per year by T & L Publications, Inc., 430 Princeland Court, Corona, CA 92879. PERIODICALS POSTAGE PAID AT CORONA, CA AND AT ADDITIONAL MAILING OFFICES. POSTMASTER: Send address changes to **Nuts & Volts, P.O. Box 15277, North Hollywood, CA 91615** or Station A, P.O. Box 54, Windsor ON N9A 6J5; cpcreturns@nutsvolts.com.

SSH ENCRYPTED SERIAL TO ETHERNET SOLUTIONS

Device P/N: SB70LC-100CR
Kit P/N: NNDK-SB70LC-KIT

$47 *Qty. 1000*

SB70LC

2-port serial-to-Ethernet server

Device P/N: SB700-EX-100CR
Kit P/N: NNDK-SB700EX-KIT

SB700EX

2-port serial-to-Ethernet server
with RS-232 & RS-485/422 support

$129 *Qty. 1000*

Device P/N: CB34-EX-100IR
Kit P/N: NNDK-CB34EX-KIT

$149 *Qty. 1000*

CB34EX

industrial temperature grade
2-port serial-to-Ethernet server
with RS-232 & RS-485/422 support
and terminal block connector

Instantly network-enable any serial device

Works out of the box - no programming is required

Customize to suit any application with low-cost development kit

256-bit encryption protects data from unauthorized monitoring

Features:

10/100 Ethernet

TCP/UDP/SSH/SSL modes

DHCP/Static IP Support

Data rates up to 921.6kbps

Web-based configuration

Need a custom solution?

NetBurner **Serial to Ethernet Development Kits** are available to customize any aspect of operation including web pages, data filtering, or custom network applications. All kits include platform hardware, ANSI C/C++ compiler, TCP/IP stack, web server, e-mail protocols, RTOS, flash file system, Eclipse IDE, debugger, cables and power supply. The NetBurner Security Suite option includes SSH v1 & v2 support.

NetBurner
Networking in One Day.

Information and Sales | sales@netburner.com
Web | www.netburner.com
Telephone | 1-800-695-6828

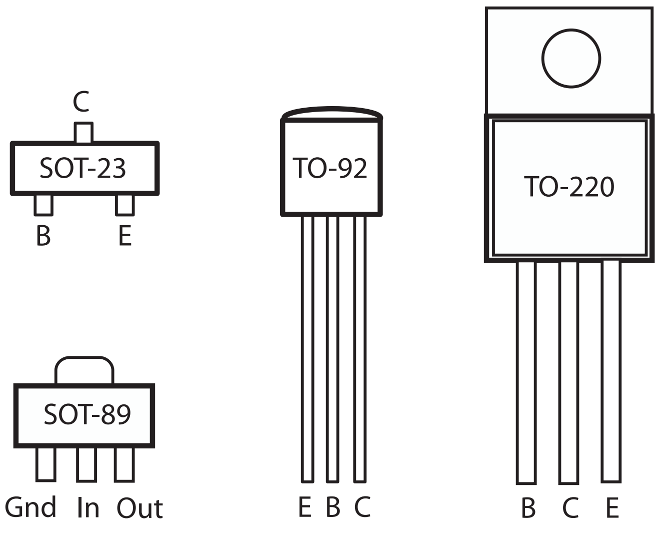
freescale™
Alliance Member

DEVELOPING PERSPECTIVES

Good Things Come With Small Packages

A good friend drops by your place and, knowing that you're an electronics enthusiast, asks for help with a new handheld gadget that suddenly stopped working. Eager to lend a hand, you check the batteries and try the reset button, but the device fails to respond. Next, you expertly pop the case and find a tiny circuit board populated with a dozen solid-state components. However, most of these components are unmarked. Moreover, a thorough search on the web fails to reveal a schematic. With your friend looking on, you suddenly feel dread and powerless to help. Sound familiar? It shouldn't and needn't be.

If we're talking about a $5 device, then the practical solution is to simply buy a new one. But sometimes there are impractical issues involved: the challenge of diagnosing and repairing a piece of electronics; the sentimental value of a device; or the need to validate your expertise in electronics to others. Whatever the reason, if you're faced with the task of repairing the circuit, don't forget to leverage the signposts staring you in the face — namely the component packaging.



Fortunately, many leaded and SMT components use standard packaging. For example, in the **diagram**, you can see that SMT transistors in the Small Outline Transistor (SOT)-23 packaging have standard lead designations. Facing the top of the package (with the single lead on top), the base lead is in the lower left lead; the emitter is the right lower lead. The collector is the lone top lead.

If you need more information — such as whether the device is PNP or NPN — it's a simple matter of using your ohmmeter to identify the polarity of the individual junctions. Recall that with an NPN transistor, the emitter-base junction conducts when the emitter is negative, relative to the base. Alternatively, you can monitor the voltages on each lead, looking for forward and reverse bias conditions to identify transistor types.

The SOT-89 package is often used with voltage regulators. As shown in the figure, ground (Gnd), input (In), and output (Out) are standard leads, from left to right. The tab on top is connected to the input lead. It's a simple matter to check the voltage input and voltages of the device to help determine whether it's working. If there's no output with an input, then the component may have failed or there may be a short in the output circuit. Many voltage regulators are short proof.

The same diagnostic walkthrough applies to transistors in leaded TO-92 and TO-220 packaging. Armed with lead designation and transistor type, a board full of components suddenly takes on new meaning. You should be able to sketch out a simplified schematic and have a good idea the area of the board responsible for the failure. All is not rosy in the land of SMT components, and there are some formidable challenges associated with the latest SMT devices, namely lack of markings because there simply isn't space on the device to write anything legible. As it is, I'm forced to rely on a 10X scope when I work with a board populated with SMT components.

Another challenge of working with SMT components is simply getting a probe on a specific lead without short-circuiting the device. To this end, I recently picked up a pair of needle-tipped probes by Fluke on eBay. I'm dreading the day when I impale myself with one of the probes, but they allow me to make measurements of SMT components that would have been impossible with standard leads.

Assuming you're successful in identifying the defective transistor or other component(s), the next step is to replace the component. For this, I rely on Digi-Key (**www.digikey.com**) and Mouser Electronics (**www.mouser.com**). A short blast of air from a hot air pencil removes the defective component from the board. If you prefer, a standard soldering iron with solder wick works just about as well.

Taking the time for shipping the replacement components into account, give yourself about a week to repair anything substantial. Not bad, when you consider most electronics devices are simply tossed in the trash at the first sign of trouble — to the detriment of the environment. **NV**

NUTS AND VOLTS
EVERYTHING FOR ELECTRONICS

Published Monthly By
T & L Publications, Inc.
430 Princeland Ct.
Corona, CA 92879-1300
(951) 371-8497
FAX **(951) 371-3052**
Webstore orders only **1-800-783-4624**
www.nutsvolts.com

Subscriptions
Toll Free **1-877-525-2539**
Outside US **1-818-487-4545**
P.O. Box 15277
North Hollywood, CA 91615

FOUNDER/ASSOCIATE PUBLISHER
Jack Lemieux

PUBLISHER
Larry Lemieux
publisher@nutsvolts.com

ASSOCIATE PUBLISHER/ VP OF SALES/MARKETING
Robin Lemieux
display@nutsvolts.com

EDITOR
Bryan Bergeron
techedit-nutsvolts@yahoo.com

CONTRIBUTING EDITORS
Jeff Eckert, Russ Kincaid, Joe Pardue, Fred Eady, Chris Savage, John Gavlik, Ron Hackett, Louis Frenzel, John Mastromoro, Thomas Kibalo

CIRCULATION DIRECTOR
Tracy Kerley
subscribe@nutsvolts.com

SHOW COORDINATOR
Audrey Lemieux

MARKETING COORDINATOR WEBSTORE
Brian Kirkpatrick
sales@nutsvolts.com

WEB CONTENT
Michael Kaudze
website@nutsvolts.com

ADMINISTRATIVE ASSISTANT
Debbie Stauffacher

PRODUCTION/GRAPHICS
Shannon Christensen

Copyright © 2010 by T & L Publications, Inc.
All Rights Reserved
All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *Nuts & Volts Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *Nuts & Volts*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *Nuts & Volts*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: **430 Princeland Court, Corona, CA 92879**.

Printed in the USA on SFI & FSC stock.

READER FEEDBACK

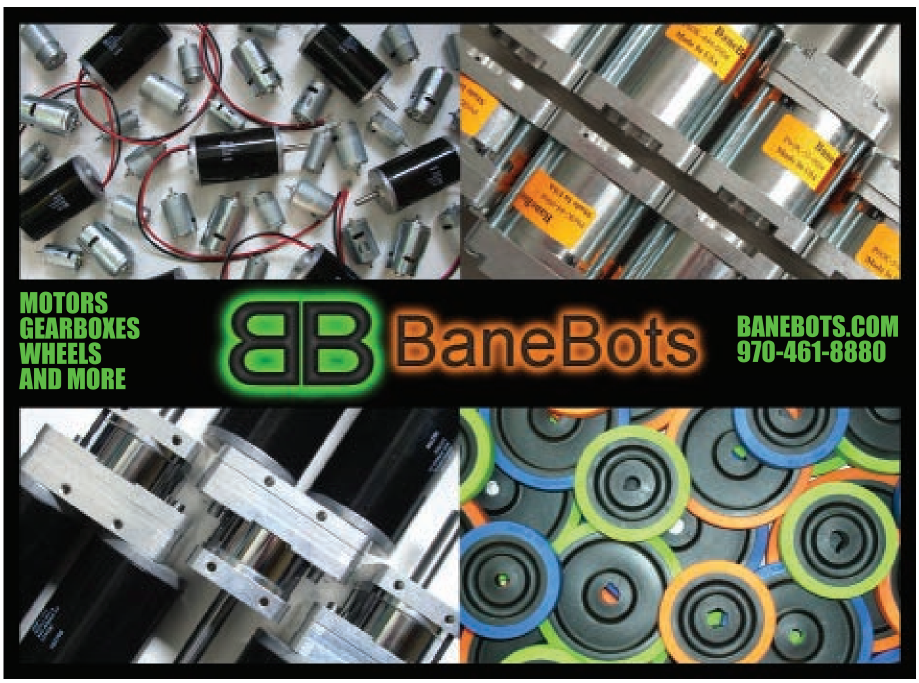
PUT A SPIN ON IT?

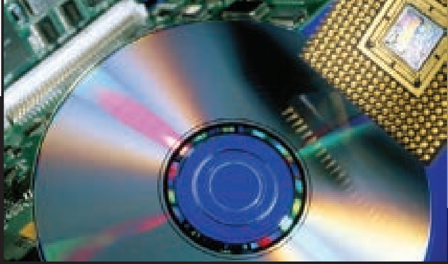
I am pretty new at electrical engineering stuff. I have been teaching myself for only about a year now (to put things in perspective). I had a quick question for Vern Graner on his December '09 Personal Robotics column regarding the Stingray. Here's a brief background. The first thing I bought was the Stampworks kit figuring that would be sufficient as a starting tool. Although I love all the cool stuff that came with it (including the development board), I found myself lost not knowing material that I should have for that kit. So I decided to get the "What's A Microcontroller?" kit which helped a lot. Although I like the BASIC Stamp, I wanted to move into the world of PICs. During this transition, I found myself wanting to learn assembly language instead of PBASIC in order to program in more detail later on (I know you think I am crazy for trying to do so much stuff in one year). I also learned how to make PCBs — single- and double-sided — starting with ferric chloride and now using the cupric chloride concoction with muriatic acid and hydrogen peroxide. Once again, I know it's a lot for a newbie but I made myself a PIC16F684 programmer with a PICkit 1 equivalent daughterboard! Anyway, the point I am making/asking is should I continue in the direction I am going, or is there a better/easier solution? I ask this because I am really interested in the Stingray, but I am unsure if I really want to learn the Propeller's language and add that to my list of kind-of learned languages or not. Do you have any suggestions for me? I apologize for boring you with my life story in electrical engineering, but I figured it would help you better understand where I am coming from.

Corey Hastings

*Welcome to the world of electronics, Corey! Fiddling with electronic devices has been a life-long preoccupation of mine and I highly recommend it. It's LOTS of fun! Seems you are REALLY attacking new challenges. That's a VERY impressive skill set you've managed to earn. I'm impressed! As for suggestions, I don't think I can answer this *objectively.* The Propeller is an amazing product in and of itself. I know of no other consumer accessible microcontroller with its power, flexibility, and support infrastructure. That being said, it is still a proprietary system and the SPIN language is unique to that chip. Learning*

Continued on page 75





TECH KNOWLEDGEY 2010

EVENTS, ADVANCES, AND NEWS

■ BY JEFF ECKERT

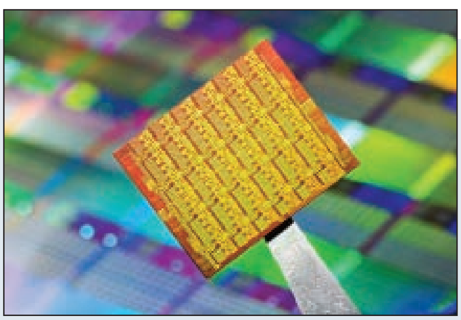
ADVANCED TECHNOLOGY

DESTRUCTION OF THE UNIVERSE BACK ON TRACK

As you probably know, the Large Hadron Collider (LCH) was created by the European Organization for Nuclear Research (CERN, **www.cern.ch**) with various goals in mind; most notably to slam opposing particle beams of either protons at an energy of 7 TeV per particle or lead nuclei at an energy of 574 TeV per nucleus, and simulate conditions that existed at the universe's kickoff. Speculation about the possible unintended consequences has included the creation of wormholes that could (a) allow time travel; (b) suck the known universe into an alternate one; or (c) release interdimensional dragons that would eat us for lunch. Another cited risk is the creation of "strangelets"—hypothetical particles that, via a chain reaction, could reduce the Earth to a hot lump of "strange matter." Fortunately, the apocalypse was postponed in Sept. 2008 when a faulty connection between two of the LCH's magnets triggered a meltdown and knocked it offline for a year. (I warned them against putting a penny in the fuseholder but, as usual, no one listened.) In any event, as of this writing, LCH has been up and running for a month or two, stable beams are circulating at 450 GeV, and collision data is being recorded. Ramping up the energy for collisions at the 7 TeV level (3.5 TeV per beam) is scheduled for early this year, so if the scientists have miscalculated, you may be reading this as you travel at the speed of light into — and simultaneously out of — the snout of a lepton-sucking quantum hog that won't exist until last year. In which case, you can expect the March issue to arrive a few months ago. Assuming you ever existed at all, you can log onto **twitter.com/cern** for the latest updates. ▲

■ Damage to the LHC magnets in sector 3-4, resulting from an "incident" in September 2008.

INTEL SHOWS 48-CORE PROCESSOR



■ Single-chip cloud computer has 48 Intel cores and draws as little as 25W.

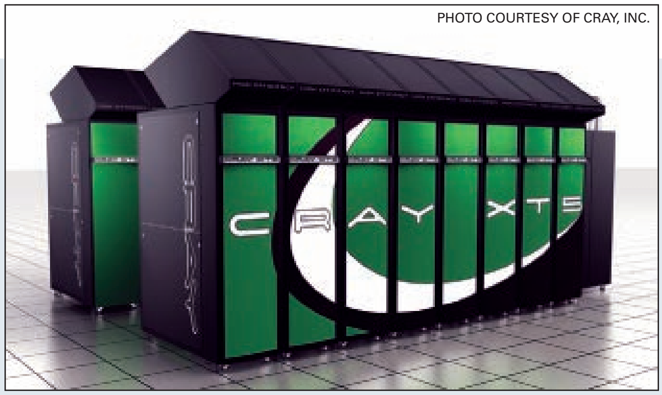
In the near term, the news from Intel (**www.intel.com**) is its upcoming line of Core-branded chips with six- and eight-core processors, due later this year. But the long-term outlook includes an experimental 48-core processor (called a "single-chip cloud computer" because it resembles the organization of data centers used to create a "cloud" of computing resources over the Internet) that is aimed at rethinking many of the designs used in present-day laptops, desktops, and servers. The company recently unveiled the prototype which contains 48 fully programmable Intel processing cores — the most ever on a single silicon chip. It also includes a high speed network and incorporates power management techniques that allow all 48 cores to operate on as little as 25W or 125 watts when running at maxed-out performance levels. Multicore programming is a well-known industry challenge, but Intel, HP, and Yahoo! Open Cirrus researchers are already porting existing cloud applications to the new chip; "Microsoft is partnering with Intel to explore new hardware and software architectures supporting next-generation client plus cloud applications." Intel plans to build 100 or more experimental chips for use by industrial and academic research collaborators around the world, so put in your request now. Details on the chip's architecture and circuits are scheduled to be revealed in a paper at the International Solid State Circuits Conference this month. You can get a CD of the conference materials at **www.isscc.org**. ▲

COMPUTERS AND NETWORKING

CRAY SLIDES PAST IBM AS TOP DOG

■ The Cray XT5 Jaguar, now the world's fastest computer.

PHOTO COURTESY OF CRAY, INC.



The first thing you notice about the Cray XT5 supercomputer is that it ain't exactly pretty. Unlike the old X-MP models which looked like retro-elegant, monolithic pieces of naugahyde lounge furniture, the new muscle machine more closely resembles a row of olive-green port-a-potties or some beer coolers at a 7-11. But, as they say, beauty's only skin deep, and beneath the tacky exterior lies a powerhouse that — in its third attempt — finally managed to knock IBM's Roadrunner from the number 1 spot on the TOP 500 (**www.top500.org**) list of supercomputing sites. Jaguar — operated by the Department of Energy's Oak Ridge Leadership Computing Facility — was upgraded last year and thereafter posted a 1.75 petaflop/s performance speed running the Linpack benchmark. Jaguar clawed its way to the top using new processors (with nearly a quarter of a million total cores) that bring its theoretical peak capability to 2.3 petaflop/s. (A petaflop/s equals 1,000,000,000,000,000 calculations per second.) As a result of a repartitioning of the system, Roadrunner actually dropped to a tawdry 1.04 petaflop/s from its previous 1.105. In the number 3 slot is another XT5 housed at the University of Tennessee, followed by an IBM BlueGene/P in Germany, and the Chinese-built Tianhe-1 (translation: river in the sky) — a hybrid design with Intel Xeon processors and AMD GPUs used as accelerators. But there's always next year. ▲

FREE LAPTOP PROTECTION

According to some widely circulated statistics, a laptop is stolen every 53 seconds; more than 12,000 of them disappear each week from US airports alone, and only three percent are ever recovered. Many software and hardware solutions are on the market — some with good success rates and some without — but few have that highly desirable quality of being free. One that is both free of charge and has received largely positive reviews is Prey: a lightweight application offered by the Prey Project (**preyproject.com**). Available for Windows, Mac OS X, and Linux, the open-source program runs in the background and periodically wakes up, goes online, and checks to see if it has been designated missing or stolen. If so, you can sign into the Prey website and follow its movements using another computer. You can also receive reports about what applications are running, files that have been modified, etc., and if the laptop has a built-in webcam, you may even get a picture of the thief. Prey will automatically check for open Wi-Fi access points, so you don't have to wait for the thief to make his own connection. If, on the other hand, Prey doesn't detect anything unusual, it just goes back to sleep. Obviously, it isn't foolproof, because a wary machine snatcher can delete it or reformat your hard drive. Still, it's better than nothing, and remember that magic word that warms the cockles of every cheapskate's heart: "free." ▲

MAC UPGRADES

PHOTO COURTESY OF APPLE.

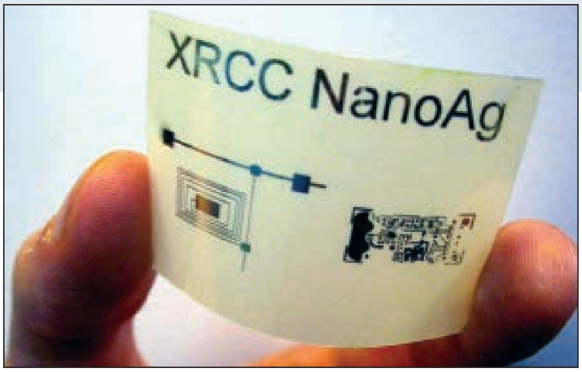
■ Newest iMacs come with 21.5 or 27 inch displays.

New desktops these days seem to be mostly evolutionary rather than revolutionary, and that includes the newest iMac models. Still, it's pretty nice that you can now get one with a 27-inch, 2560 x 1440 pixel widescreen display that's suitable for watching high-def movies and TV shows. Both come with a wireless keyboard and Magic Mouse which replaces mechanical buttons, scroll wheels, and scroll balls with a seamless multi-touch surface that allows a user to scroll through documents, pan across large images, or swipe to move forward or backward through web pages or photos. The Magic Mouse can be configured as either a single- or two-button mouse, thus resolving a long-running dispute between traditional Mac users and those who are more accustomed to PC-style input devices. The machines start at $1,199 for a 21.5 inch model with a 3.06 GHz Core 2 Duo processor and ratchet up to $1,999 if you want the big screen and a quad-core processor. If you bump the 27-incher up to 16 GB of RAM and a 2 TB drive, you're suddenly at $3,649, so frills don't come cheap. You can test price your own configuration at **store.apple.com/us**. ▲

CIRCUITS AND DEVICES

NEW INK ENABLES CHEAP ELECTRONICS

If you're looking for a cheap way to add intelligence and/or computing power to plastics, fabrics, and other flexible surfaces, check out the new low temperature silver ink developed at the Xerox Research Centre of Canada. "For years, there's been a global race to find a low cost way to manufacture plastic circuits," said Paul Smith, laboratory manager. "We've found the silver bullet that could make things like electronic clothing and inexpensive games a reality today. This breakthrough means the industry now has the capability to print electronics on a wider range of materials and at a lower cost."



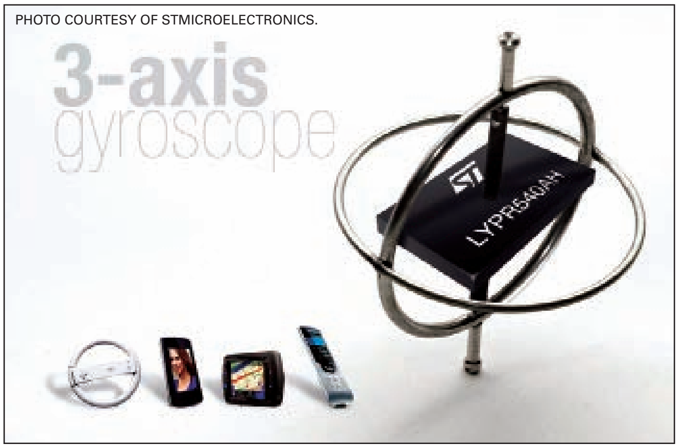
With the new process, circuits can be printed like any continuous-feed document without the need for clean room facilities or other exotic setups. In addition, the ink has been improved such that the molecules align themselves in the best configuration to conduct electricity. Cited product applications include low cost RFID tags; lightweight, flexible e-readers; signage; sensors; solar cells; and novelty applications including wearable electronics. Interested parties are encouraged to contact Xerox for samples of the materials, including the new ink. No specific contacts were offered, but you can always start at **www.xerox.com**. ▲

THREE-AXIS GYROSCOPE IN A SINGLE PACKAGE

A pretty nifty gadget recently introduced by STMicroelectronics is the LYPR540AH three-axis analog gyroscope which — in a single package — measures angular rates along three orthogonal axes. The sensor offers 360° detection for high precision 3D gesture and motion recognition in mobile phones, game controllers, personal navigation systems, and other portable devices. If you combine it with a three-axis accelerometer (and ST claims to have sold more than 600 million of them), the device enables the creation of inertial measurement units (devices that track and deliver information on the type, rate, and direction of motion). The gyro offers two outputs for each axis: a 400-dps full-scale value for high accuracy of slow motion and a 1,600-dps full-scale value for fast movements. They operate in a temperature range from -40 to 85°C and measure only 4.4 x 7.5 x 1.1 mm. Samples are available, and the devices are priced at $3.60 in manufacturing quantities. **NV**

■ Three-axis MEMS gyroscope.

PHOTO COURTESY OF STMICROELECTRONICS.



■ BY LOUIS E. FRENZEL W5LEF

WHAT'S NEW IN HOME NETWORKING?

It's not all wireless as a new wired home standard G.hn has arrived.

Who would have thought we would all need a communications network in our homes? These days, some kind of network is almost a necessity with more members of the family using a PC or notebook. Also, we've begun to rely on the Internet as our source of TV and other entertainment.

The Internet has become our main source of information and outside world connectivity. Over 70% of US homes now report that they have a high speed Internet service. What this means is that we need a distribution system that connects that single Internet link to multiple computers and, in some cases, our home entertainment systems and even telephones. Thus, the home network.

WIRELESS IS KING OF THE CASTLE

The home network is a way to connect your DSL line, cable modem, or wireless modem bringing the Internet to all the devices that need it. Up to now, that has meant connecting to one or more PCs that are located in different places around the house. You can implement an Ethernet LAN in your home, but who wants to run CAT5 or CAT6 twisted pair cable all round the house?

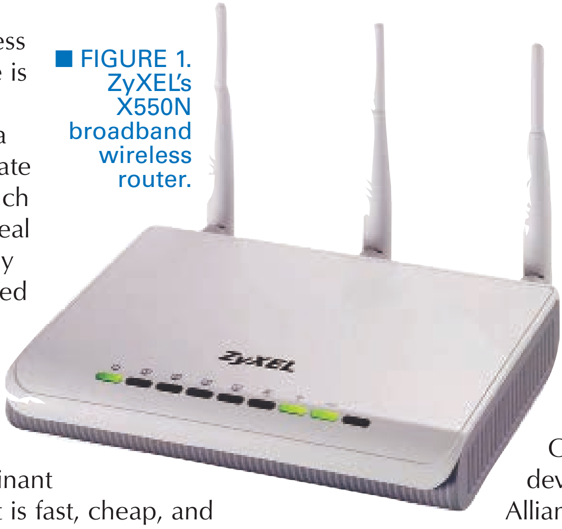
As a result, the most popular way to implement a home network is through a simple wireless network using Wi-Fi — a widely accepted wireless networking standard and system. It is fast, simple, and cheap. All you do is buy a wireless router and connect it to your DSL, cable, or wireless modem, and it will link up with the wireless transceivers already built into most PCs and laptops. It will even connect to Wi-Fi enabled smartphones.

In case you don't know, Wi-Fi is the popular name for wireless networking gear that uses the IEEE's 802.11 radio standards. That standard has been around since 1997 and it has evolved from a costly low speed wireless connection to a ubiquitous low cost wireless technology. It is the technology you use when you connect to a wireless hot spot in an airport, hotel, or other public place.

The first widely successful wireless LAN radios had the designation 802.11b. This uses the 2.4 GHz unlicensed frequency band and can deliver a maximum data rate of 11 Mb/s (million bits per second) up to a maximum range of 100 meters. The data rate automatically ratchets down to 5.5, 2, or 1 Mb/s depending on the range, noise, and any interference or obstructions (like walls and trees). It works great, actually. An improved standard called 802.11g came along next, and provided a data rate up to 54 Mb/s. It too adjusted downward in speed to maintain contact under varying locations and conditions. Another available standard is 802.11a. It also offers data rates to 54 Mb/s, but uses the 5.8 GHz spectrum. The maximum range is a bit shorter but there is less interference from other radios like cordless phones, Bluetooth headsets, microwave ovens, and other electromagnetic radiators in the popular 2.4 GHz band.

The latest version of the standard is 802.11n. After many years of development, products using this standard are now available. Depending on its configuration, this standard can deliver data rates over 100 Mb/s and, in some cases, from 300 to 600 Mb/s. This is possible because the radios use MIMO or multiple input multiple output. MIMO is a technique of using multiple transmitters, receivers, and antennas to boost the data rate, as well as provide longer range and a more reliable connection. MIMO transmits parallel data streams on the same frequency using different antennas spaced from one another. The receiving end uses multiple antennas and receivers to pick up the signals from different paths. A complex digital signal processing technique sorts all the signals out and combines them back into the original single high speed data stream.

A good example of a wireless router for home networking use is the ZyXEL X550N shown in **Figure 1**. It uses a three-antenna MIMO that can deliver a data rate to 300 MB/s. That is usually much faster than anyone needs. The real value of the MIMO is the greatly extended range and the increased reliability of connectivity. It is backwards-compatible with 802.11b/g radios, so older gear can be connected without trouble.



■ FIGURE 1. ZyXEL's X550N broadband wireless router.

Wi-Fi has become the dominant home networking technology. It is fast, cheap, and best of all, it requires no wires. Thanks to the rigid testing of the Wi-Fi Alliance, all Wi-Fi products (regardless of manufacturer) are fully compatibility and interoperable with one another. It is a no-brainer if you haven't set up a home net in your house yet.

WIRING IS NOT DEAD

While wireless is still the first choice of most consumers, there are a few good wired systems out there. They do not require you to add any new wiring but instead use the AC power line, the internal telephone wiring, or your home coax cable TV wiring for communications. AC power line networking has been around for years, and is widely used by electrical utilities for transmitting data over the power lines. In the 1990s, several companies created power line data communication modems for home use.

The idea is to superimpose a modulated data stream on the regular 120 volt AC mains. You can inject it at any AC outlet and it will be available at any other outlet as it makes its way through the house wiring. It's a neat idea that works pretty well. However, you do have to fight the high noise level that exists on the power line. This is overcome in most versions of the technology by using orthogonal frequency division multiplexing (OFDM). OFDM is the technique of dividing a high speed serial data stream into hundreds of parallel slower streams and modulating each one on adjacent channels. This is done by using digital signal processing. The technique is similar to that used in DSL lines where the OFDM technique is called DMT or discrete multitone (DMT). It not only provides the high speed, but the technique is highly immune to noise.

A number of different power line networking standards have been developed over the years using this approach. All use some variation of OFDM. The most widely used was developed by the Home Plug Powerline Alliance (HPPA); **www.homeplug.org**. Their most recent version of the standard is HomePlug AV (audio/video). It can deliver a data rate of up to 200 Mb/s for networking video, audio, and computer data.

Another power line standard is the High-Definition Power Line Communications Alliance (HD-PLCA); **www.hd-plc.org**. HD-PLCA is used in the US and Japan. It can produce data rates to 210 Mb/s. The Universal Powerline Association (UPA; **www.upaplc.org)** is the source of another power line standard. Their objective was to create a worldwide standard for this technology. The IEEE has their own standard for power line networking called P1901. It defines two versions of OFDM to be compatible with other standards. It was an attempt to create a single worldwide standard, but has not been widely adopted.

While AC power line routers are available for home networking, you can never be sure which standard it uses. If you stick with one manufacturer and standard, then all your connected devices will talk to each other. You cannot mix and match the different standard types as they are incompatible. That has confused some consumers and made others abandon the method. As a result, none of the various standards or products have ever emerged as a clear winner. On top of that, several other methods have been created to do the same thing.

For example, the HomePNA Alliance (HPNA; **www.homepna.org**) standard uses either the twisted pair telephone wiring or coax cable — both buried in most homes today. It can achieve 320 Mb/s using a DSL-like technique. It is used in AT&T's U-verse IPTV distribution system. There is also the Multimedia over Coax Alliance (MoCA; **www.mocalliance.org**) standard that uses the cable TV coax. It transmits the data backwards through the splitters to all wall outlets with F-connectors. The maximum data rate is 175 Mb/s. Verizon's FiOS fiber optic broadband network uses MoCA.

All of these standards — except HPNA and MoCA — use the AC power wiring in a home as the transmission medium and all have products in the field. The physical layer (PHY) technology in all cases is some variation of

■ FIGURE 2. The potential of the G.hn home networking standard lies in four major areas including: consumer electronic connectivity; PC and peripherals networking; service provider connectivity, and smart grid communications.

OFDM, including both FFT and wavelet processes. This is the only modulation method proven to stand up to the noisy AC line. HPNA and MoCA use the cable TV coax that exists in most homes. It too uses OFDM.

Collectively, all these standards have never achieved the volume level of wireless home networking with Wi-Fi. The lack of a single compatible standard is considered to be the problem. So, the industry has set out to create a worldwide wired home networking standard that is referred to as G.hn.

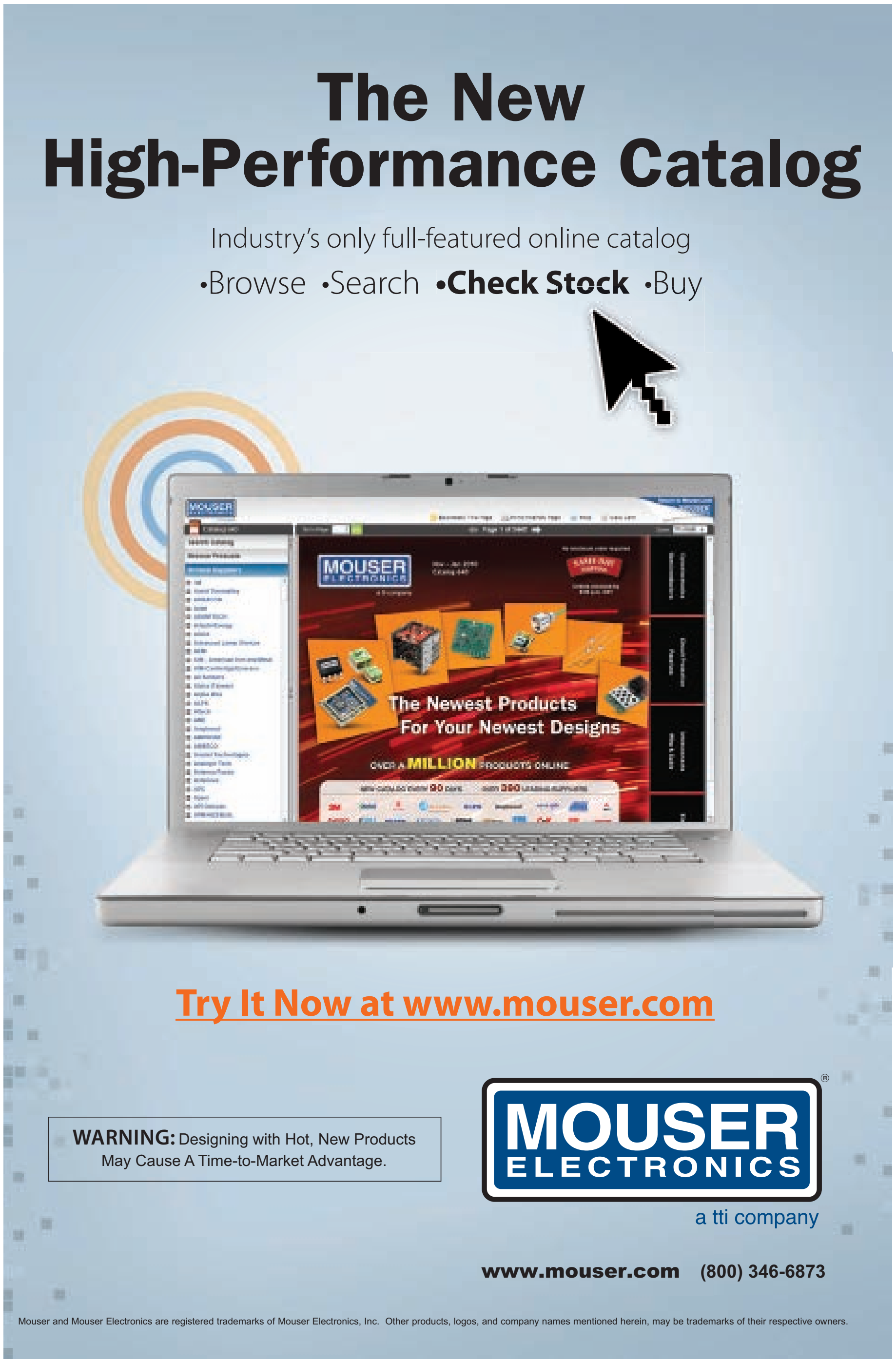
WELCOME TO G.HN

G.hn is the short hand designation for the International Telecommunications Union's (ITU) attempt to create one world standard that could serve everyone. Furthermore, the goal was to produce a networking technology that would be used not only to link home PCs to the Internet, but also could serve to connect home entertainment devices like HDTV sets and DVD players, as well as be the connection method of choice for Internet service providers (like cable companies with their set top boxes). On top of that, they wanted a standard that could be used to connect the home electrical system and appliances for monitoring and control as proposed by the so-called "smart grid" green effort by many utilities. **Figure 2** shows the vision of the HomeGrid Forum — an organization that supports and promotes the G.hn standard.

The G.hn standard has been in the works since about 2006 but it finally emerged from a long development period in October last year. It is now complete enough where the chip companies can begin working on ICs to implement the modems. The formal designation is G.9960 and its companion standard G.9972 offers up a co-existence mechanism that allows G.hn devices to work with devices using other standards. Their overall goal was to come up with a technology that would work over coax, twisted pair, or the AC line with a common protocol and networking mechanism making one compatible standard.

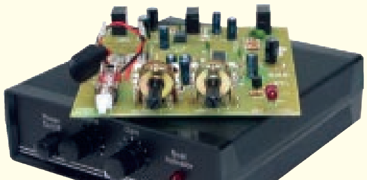
Like all the other standards, G.hn uses OFDM that operates over the 2 to 30 MHz range on the wired medium. Initially, it can support data rates of over 100 Mb/s with a promise of 1 Gb/s in later versions. It also used a forward error correction (FEC) method called the quasi-cyclic low density parity check (QC-LDPC) to improve transmission reliability. It has all sorts of other features that make it a quality, high-end wired networking method. For more details, go to **www.homegridforum.org**.

There are no chips or end products yet, but these are in the works. It will be interesting to see what products emerge. Or, will the existing standards continue with the potpourri of different incompatible standards? We shall see. **NV**

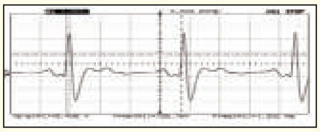


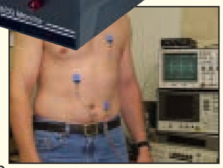
Valentine's Day Goodies!

Electrocardiogram ECG Heart Monitor



- ✔ **Visible and audible display of your heart rhythm!**
- ✔ **Bright LED "Beat" indicator for easy viewing!**
- ✔ **Re-usable hospital grade sensors included!**
- ✔ **Monitor output for professional scope display**
- ✔ **Simple and safe 9V battery operation**





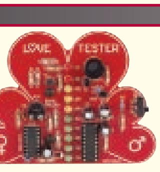
February is the month for Valentine's Day, and what a great time to think of your heart! Not how many times it's been broken, not how many times it's fallen head over heels in love, but how it actually works... and how it's doing these days!

Use the ECG1C to astound your physician with your knowledge of ECG/EKG systems. Enjoy learning about the inner workings of the heart while, at the same time, covering the stage-by-stage electronic circuit theory used in the kit to monitor it. The three probe wire pick-ups allow for easy application and experimentation without the cumbersome harness normally associated with ECG monitors. The documentation with ECG1C covers everything from the circuit description of the kit to the circuit description of the heart! Multiple "beat" indicators include a bright front panel LED that flashes with the actions of the heart along with an adjustable level audio speaker output that supports both mono and stereo hook-ups. In addition a monitor output is provided to connect to any standard oscilloscope to view the traditional style ECG/EKG waveforms just like you see on ER... or in the ER! See the display above? That's one of our engineers, hooked up to the ECG1C after an engineering meeting!

The fully adjustable gain control on the front panel allows the user to custom tune the differential signal picked up by the probes giving you a perfect reading and display every time! 10 hospital grade re-usable probe patches are included together with the matching custom case set shown. Additional patches are available in 10-packs. Operates on a standard 9VDC battery (not included) for safe and simple operation. Note, while the ECG1C professionally monitors and displays your heart rhythms and functions, it is intended for hobbyist usage only. If you experience any cardiac symptoms, seek proper medical help immediately!

ECG1C	**Electrocardiogram Heart Monitor Kit With Case & Patches**	**$44.95**
ECG1WT	**Electrocardiogram Heart Monitor, Factory Assembled & Tested**	**$89.95**
ECGP10	**Electrocardiogram Re-Usable Probe Patches, 10-Pack**	**$7.95**

Electronic "Love Tester"



- ✔ **10 LED love scale display!**
- ✔ **Audible love level sound!**
- ✔ **Great party fun!**
- ✔ **Heart shaped PC board!**

This uniquely shaped "Love Tester" is the ultimate gag for any couple! Designed to check your love life, each partner holds one end of the tester PCB at the appropriate male and female touch pads. Then they romantically join hands and watch the results on the love meter! 10 green, yellow, and red LEDs act like a scale, and just like the carnival when it hits the top they flash, indicating you're a red hot couple! There is also an audible alarm that changes with the "love level". Next time the party isn't going anywhere, bring this out, it's a riot!

Wide sensitivity range is compatible with all couples. Includes a built-in power on/off switch. Runs on a standard 9V battery. Measures 4.1" x 3.1" x .98".

MK149	**Electronic Love Tester Kit**	**$16.95**

LED Flashing Sweetheart

- ✔ **28 brilliant red LEDs!**
- ✔ **Unique dual heart design!**
- ✔ **Freestanding mount!**

What a way to display your feelings to that very special person in your life! Get out your soldering iron and Dazzle her with this unique dual heart electronic display that you can say you built yourself!

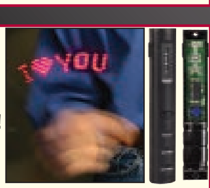
28 brilliant red LEDs are formed into two separate heart designs on the heart shaped PC board creating a great flashing display that's easy to assemble.

Built-in battery holder base makes the sweetheart freestanding and perfect for a desk or table.

Runs on a standard 9V battery (not included). Measures 2.4" x 2.4" x 1.2".

MK101	**LED Flashing Sweetheart Kit**	**$6.95**

LED Magic Message Wand



- ✔ **Message floats in air!**
- ✔ **Fun at concerts & events!**
- ✔ **High visibility red LEDs**
- ✔ **Pre-programmed or custom messages**

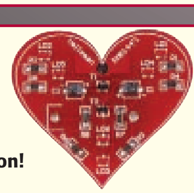
Use the "Magic Wand" to display your true feelings! Simply shake it back and forth and brilliant messages seem to appear in mid-air! Six high intensity LEDs are microprocessor controlled to display messages and graphics that are pre-programmed into the wand.

You can also custom program a message of your choice! From amazing your friends, making a statement at a concert, or simply telling your loved one how you feel, the message wand can't be beat!

Runs on two AAA batteries (not included), and the auto power off circuit gives you long battery life.

MK155	**LED Magic Wand Kit**	**$12.95**

SMT LED Flashing Heart Display



- ✔ **Alternating flashing!**
- ✔ **6 super bright SMT LEDs!**
- ✔ **Learn all about SMT!**
- ✔ **Definitely gets her attention!**

This cute little kit gives you a distinctive red display using 6 Surface Mount (SMT) LEDs. The PC board is in the shape of a red heart. The small size makes it perfect to be used as a badge or hanging pendant around your neck. Even better as an illuminated attention-getting heart to accompany a Valentines Day card!

Makes a great SMT learning kit to bring you into the world of SMT technology, design, and hands-on soldering and troubleshooting. Don't worry, extra SMT parts are included just in case you lose or damage any! Runs on a small CR2025 or CR2032 button cell (not included). Measures 1.9" x 1.7" x .3".

MK144	**SMT LED Flashing Heart Kit**	**$11.95**

Our Best Seller Favorites!

FM Stereo Transmitter

- ✔ **Rock stable PLL synthesized exciter**
- ✔ **Front panel digital control and display of all parameters!**
- ✔ **Professional metal case**
- ✔ **Super audio quality!**
- ✔ **25mW kit and 1W export models!**

For nearly a decade we've been the leader in hobbyist FM radio transmitters. When it became time for a new model, we started from the ground up! We told our engineers we wanted a new technology transmitter that would provide FM100 series quality without the advanced mixer features. They took it as a challenge and designed not one, but TWO transmitters!

The FM30 is designed using through-hole technology and components and is available only as a do-it-yourself kit, with a 25mW output very similar to our FM25 series. Then the engineers redesigned their brand-new design using surface mount technology (SMT) for a very special factory assembled and tested FM35WT version, with 1W output for our export-only market! Both are designed around an RF tight vinyl clad metal enclosure for noise free and interference free operation. All settings are done through the front panel digital control and LCD display! All settings are stored in non-volatile memory for future use.

Both the FM30 and FM35WT operate on 13.8 to 16VDC and include a 15VDC 110/220VAC plug in power supply. The stylish black anodized aluminum case measures 5.55"W x 6.45"D x 1.5"H. and is a great match to your other equipment. *(Note: After assembly of this do-it-yourself hobby kit, the user is responsible for complying with all FCC rules & regulations within the US, or any regulations of their respective governing body. FM35BWT is for export use and can only be shipped to locations outside the continental US or valid APO/FPO addresses or valid customs brokers for end delivery outside the continental US.)*

FM30B	**FM Stereo Transmitter Kit**	**$199.95**
FM35BWT	**1W Export Only Version**	**$299.95**

Electronic Learning Labs

- ✔ **Learn and build!**
- ✔ **130, 200, 300, & 500 in one!**
- ✔ **Super comprehensive training manuals!**

Whether you want to learn the basics of electricity, the theory of electronics, or advanced digital technology, our lab kits are for you! Starting with our PL130, we give you 130 different electronic projects, together with a comprehensive 162 page learning manual. A great start for the kids...young and old! Next, check out the PL200, that gives you 200 very creative and fun projects, and includes a neat interactive front panel with 2 controls, speaker, LED display and a meter.

From there, step up to our PL300, which gives you 300 separate electronic projects along with 165 page learning and theory manual. The PL300 walks you through the learning phase of digital electronics.

If you're looking for the ultimate lab kit, check out our PL500. Includes a whopping 500 separate projects, a 152 page starter course manual, a 78 page advanced course manual, and a 140 page programming course manual! The PL500 covers everything from the basics to digital programming! Learn all about electronics theory, and digital technology the fun way and build some neat projects!

PL130	**130-In-One Lab Kit**	**$49.95**
PL200	**200-In-One Lab Kit**	**$69.95**
PL300	**300-In-One Lab Kit**	**$89.95**
PL500	**500-In-One Lab Kit**	**$219.95**

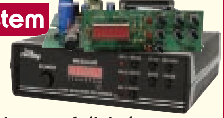
Vintage Battery Eliminator

Collectors come across some great deals on antique battery-powered radios, but how to power them is a real problem. Many classic radios operated on batteries only, and in many cases a series of three batteries for each radio were required!

The new ABCE1 Battery Eliminator gives you an easy way to replace all these batteries with a simple household AC power connection and resurrect your vintage antique radios! Provides "A" filiment, "B" plate, and "C" control grid supplies, which are all isolated from each other. Complete with aluminum case. Runs on 110-240VAC.

ABCE1 Vintage Radio Battery Elim Kit $199.95

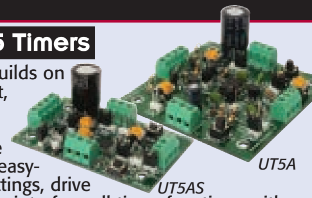
Digital Message System



The third generation of Ramsey digital voice storage kits! We started with the latest digital voice storage technology. It provides up to 8 minutes of digital storage at a frequency response up to 3.5 KHz. (Total message time and frequency response is dependant on selected internal sampling rate.) Once recorded, messages are available for playback on-demand or automatic continuous looping. Standard RCA unbalanced line level output is provided for easy connection to any amplifier, amplified speaker, mixer, or sound system. In addition, a standard 4-8 ohm speaker output is provided to directly drive a monitor speaker. Can be remote controlled via 3-wire BCD with our interface options. Check www.ramseykits.com for all options!

DVMS8 Digital Voice Message 8Ch Kit $99.95
DVMS8WT Assembled DVMS8 $149.95

Ultimate 555 Timers

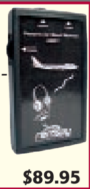


This new series builds on the classic UT5 kit, but takes it to a whole new level! You can configure it on the fly with easy-to-use jumper settings, drive relays, and directly interface all timer functions with onboard controls or external signals.

All connections are easily made though terminal blocks. Plus, we've replaced the ceramic capacitor of other timer kits with a Mylar capacitor which keeps your timings stable over a much wider range of voltages! Available in through hole or surface mount versions! Visit www.ramseykits.com for version details.

UT5A Through Hole 555 Timer/Osc Kit $24.95
UT5AS SMT 555 Timer/Osc Kit $26.95

Passive Aircraft Monitor



The hit of the decade! Our patented receiver hears the entire aircraft band without any tuning! Passive design has no LO, therefore can be used on board aircraft! Perfect for airshows, hears the active traffic as it happens! Available kit or factory assembled.

ABM1 Passive Aircraft Receiver Kit $89.95

Voice Activated Switch

Voice activated (VOX) provides a switched output when it hears a sound. Great for a hands free PTT switch or to turn on a recorder or light! Directly switches relays or low voltage loads up to 100mA. Runs on 6-12 VDC.

VS1 Voice Switch Kit $9.95

OBDII *CarChip Pro*

The incredible OBDII plug-in monitor that has everyone talking! Once plugged into your vehicle it monitors up to 300 hours of trip data, from speed, braking, acceleration, RPM and a whole lot more. Reads and resets your check engine light, and more!

8226 CarChip Pro OBDII Monitor-Asmb $99.95

RF Preamplifier

The famous RF preamp that's been written up in the radio & electronics magazines! This super broadband preamp covers 100 KHz to 1000 MHz! Unconditionally stable gain is greater than 16dB while noise is less than 4dB! 50-75 ohm input. Runs on 12-15 VDC.

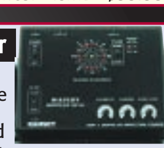
SA7 RF Preamp Kit $19.95

Touch Switch

Touch on, touch off, or momentary touch hold, it's your choice with this little kit! Uses CMOS technology. Actually includes TWO totally separate touch circuits on the board! Drives any low voltage load up to 100mA. Runs on 6-12 VDC.

TS1 Touch Switch Kit $9.95

Doppler Direction Finder



Track down jammers and hidden transmitters with ease! 22.5 degree bearing indicator with adjustable damping, phase inversion, scan and more. Includes 5 piece antenna kit. Runs on 12VDC vehicle or battery power.

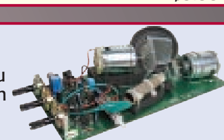
DDF1 Doppler Direction Finder Kit $169.95

Mad Blaster Warble Alarm

If you need to simply get attention, the "Mad Blaster" is the answer, producing a LOUD ear shattering raucous racket! Super for car and home alarms as well. Drives any speaker. Runs on 9-12VDC.

MB1 Mad Blaster Warble Alarm Kit $9.95

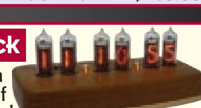
Laser Light Show



Just like the big concerts, you can impress your friends with your own laser light show! Audio input modulates the laser display to your favorite music! Adjustable pattern & speed. Runs on 6-12VDC.

LLS1 Laser Light Show Kit $49.95

Retro Nixie Tube Clock



Genuine Nixie tubes popular in the 50's brought back in one of the neatest digital clocks around today! Hand made teak maple base, 12/24 hour format, soft fade-out, auto-dim, and a crystal time base at 20ppm! Tube kits also available.

IN14TM Teak Maple Nixie Clock Kit $329.95

Water Sensor Alarm

This little $7 kit can really "bail you out"! Simply mount the alarm where you want to detect water level problems (sump pump)! When the water touches the contacts the alarm goes off! Sensor can even be remotely located. Runs on a standard 9V battery.

MK108 Water Sensor Alarm Kit $6.95

USB DMX Interface

Control DMX fixtures with your PC via USB! Controls up to 512 DMX channels each with 256 different levels! Uses standard XLR cables. Multiple fixtures can be simply daisy chained. Includes Light Player software for easy control. Runs on USB or 9V power.

K8062 USB DMX Interface Controller Kit $67.95

HV Plasma Generator

Generate 2" sparks to a handheld screwdriver! Light fluorescent tubes without wires! This plasma generator creates up to 25kV at 20kHz from a solid state circuit! Build plasma bulbs from regular bulbs and more! Runs on 16VAC or 5-24VDC.

PG13 HV Plasma Generator Kit $64.95

Air Blasting Ion Generator

Generates negative ions along with a hefty blast of fresh air, all without any noise! The steady state DC voltage generates 7.5kV DC negative at 400uA, and that's LOTS of ions! Includes 7 wind tubes for max air! Runs on 12-15VDC.

IG7 Ion Generator Kit $64.95

Tickle-Stick Shocker

The kit has a pulsing 80 volt tickle output and a mischievous blinking LED. And who can resist a blinking light and an unlabeled switch! Great fun for your desk, "Hey, I told you not to touch!" Runs on 3-6 VDC.

TS4 Tickle Stick Kit $12.95

Speedy Speed Radar Gun

Our famous *Speedy* radar gun teaches you doppler effect the fun way! Digital readout displays in MPH, KPH, or FPS. You supply two coffee cans! Runs on 12VDC or our AC125 supply.

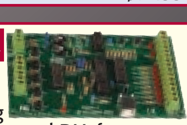
SG7 Speed Radar Gun Kit $69.95

Tri-Field Meter Kit

"See" electrical, magnetic, and RF fields as a graphical LED display on the front panel! Use it to detect these fields in your house, find RF sources, you name it. Featured on CBS's Ghost Whisperer to detect the presence of ghosts! Req's 4 AAA batteries.

TFM3 Tri-Field Meter Kit $74.95

USB Experimenter's Kit



Get hands-on experience developing USB interfaces! 5 digital inputs, 8 digital outputs, 2 analog I/O's! Includes diagnostic software and DLL for use with Windows based systems. The mystery is solved with this kit!

K8055 USB Experimenter's Kit $49.95

3-In-1 Multifunction Lab

The handiest item for your bench! Includes a RoHS compliant temp controlled soldering station, digital multimeter, and a regulated lab power supply! All in one small unit for your bench! It can't be beat!

LAB1U 3-In1 Multifunction Solder Lab $129.95

Just Off The Press!

Get the new 2010 Ramsey Hobby Catalog! ***96 value packed pages*** of the neatest goodies around with lots of new stuff! Order yours today on line or give us a call... Or download the PDF at www.ramseykits.com/catalog!

www.ramseykits.com

ramsey
www.ramseykits.com

590 Fishers Station Drive
Victor, NY 14564
(800) 446-2295
(585) 924-4560

800-446-2295

Where Electronics Is Always Fun!

✦ ***Build It!*** ✦ ***Achieve It!***
✦ ***Learn It!*** ✦ ***Enjoy It!***

Prices, availability, and specifications are subject to change. According to Robin we are not responsible for typos, stupids, printer's bleed, or broken hearts. Possession of this Valentine's Day ad will not guarantee you a date! Visit www.ramseykits.com for the latest pricing, specials, terms and conditions. Copyright 2010 Ramsey Electronics, LLC...so there!

Q&A

WHAT'S UP:

Join us as we delve into the basics of electronics as applied to every day problems, like:

- ✔ Count Down Timer
- ✔ Two-Sided PCBs
- ✔ Large Clock

■ WITH RUSSELL KINCAID

In this column, I answer questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions.

Send all questions and comments to:
Q&A@nutsvolts.com

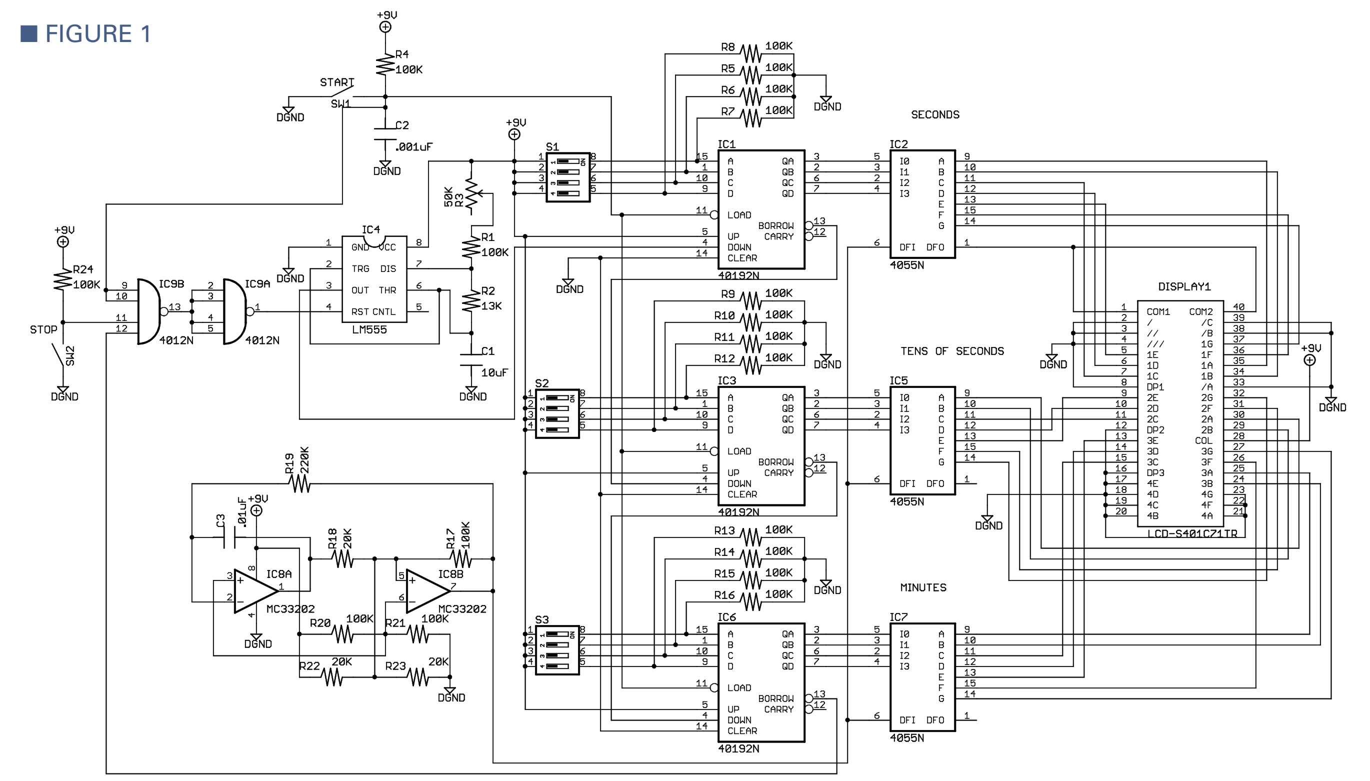
COUNT DOWN TIMER

Q We play a lot of board games at the kitchen table. I would like to build a digital timer to count down from, say 3:00 or 3:30 minutes to 0:00, then stop. A buzzer or LED would then tell that time is up. I was thinking of three buttons: a set button, a start button, and a stop button, plus DIP switches to set the start time. It would need a timing circuit like an LM555 chip. If the display shows 3:00 at start, the next display would be 2:59, 2:58, 2:57, and so on down to 2:00 and then 1:59, 1:58, down to 0:00, then stop. I want to be able to stop the timer if a person makes a word before the end, and be able to reset it. I am not into PIC programming yet, so could you show me a circuit using discrete parts and chips? The power supply will be a nine-volt battery.

— Robert Cuneo

A The circuit of **Figure 1** is rather complex, but I admit that I designed it faster than I could have written a program to do this job. A nine-volt battery does not have a lot of power so I used CMOS chips and a liquid crystal display. I had not used that type of display before, so I breadboarded it to be sure I understood how it works. The DIP switches — S1, S2, and S3 — are used to set the start time. The switches are weighted: 1, 2, 4, and 8, so for three

■ FIGURE 1

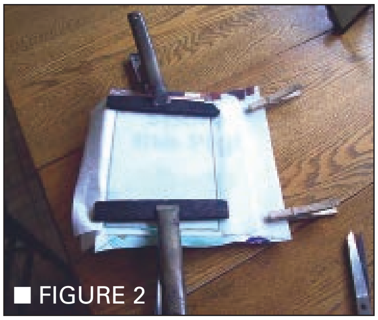


seconds, you would turn on 1 and 2 of SW1. For nine seconds, turn on 4 and 1 of SW1. The 40192 is a BCD counter which means it stops at nine counts; a binary counter (40193) would stop at 15. The maximum start time that you can set is nine minutes 99 seconds. Actually, you can set the start time to be 15 minutes, 15 seconds but this is outside the capability of the counter and I am not going to predict what will happen (it is predictable, however).

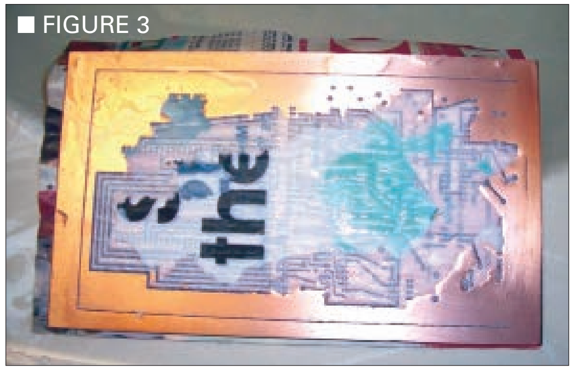
When the counter output goes to zero, the borrow output goes low then goes high on the next clock pulse which clocks the next counter. When the minutes counter reaches zero, the borrow output goes low and stops the clock. It remains in this condition until the start button is pushed. When the start button is pushed, the start time is loaded into the counters and all the borrow outputs go high. When the start button is released, the clock starts and the timer counts down. IC4 is a 1 Hz clock. A trimpot allows you to adjust the frequency to compensate for component tolerance. SW2 should be a toggle or latching pushbutton so the clock stays stopped until released.

IC8 generates a symmetrical square wave for the LCD display. The 4055 produces in phase or out of phase signals to determine which segments are displayed. I did not know if unused inputs should be left open or grounded, so I did some of both. I will fix it when I find out which is correct.

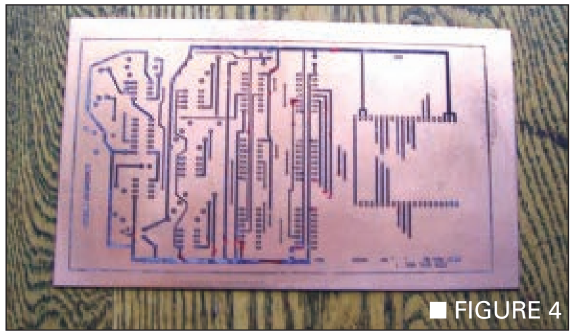
There have not been many questions this month, so I decided to show how I make a two-sided PCB (printed circuit board). Since this circuit is an ideal candidate. In order to minimize crossovers, I run power and ground up opposite sides of the board. Power and ground runs will be horizontal, while signal runs are vertical. Signals may run horizontal if necessary. The layout program that I use (Eagle) is designed for plated thru holes, so I increase the diameter of vias to allow for some registration error. I have been using PNP blue but it seems to deteriorate with time, so I have gone back to using glossy paper. My newspaper often comes with 8-½ x 11 glossy inserts; I save them for making PCBs. (Kinko's carries glossy paper which can be purchased.) The process is: Print the pattern on glossy paper using a laser printer (or copier). Don't forget to mirror the top layer. Hold the two papers pattern to pattern over a strong light (I use a sun lit window) and line up the patterns as perfectly as you can. Hold the papers in place with clothespins or clamping paper clips. There must be an inch or more of paper beyond the pattern so you can insert the double-sided copper-clad without disturbing the pattern registration; see **Figure 2**.



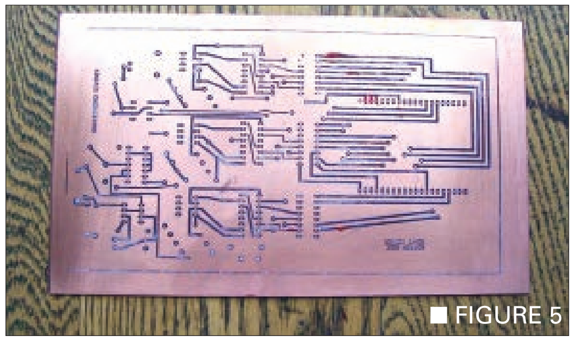
■ FIGURE 2



■ FIGURE 3



■ FIGURE 4



■ FIGURE 5

Now I clamp the paper and copper-clad sandwich between two pieces of glass and heat in the oven at 440- 450 degrees F. I get good results by putting a layer of tissue paper (Kleenex) over the paper on both sides. I use strips of wood to protect the glass from the metal clamps. I set the oven on preheat and as soon as it is up to temperature, I shut it off and let it cool gradually. Otherwise, the glass will crack. As soon as it is cool enough to handle, I put the paper and copper-clad in warm water until the paper is soaked. The paper sticks to the pattern pretty good, so you may have to use a fingernail brush to

MAILBAG

Dear Russell:

Re: November issue, page 29. In Figure 1, your reader suggested using an 8051. Since I am trying to "spread the word," I would like to suggest he find out about the PICAXE chip. Easy to program in BASIC and very cheap. Instead of taking hours, it only takes minutes to program. Like Geico says, "A caveman could do it!" I am the self proclaimed leader of the most PICAXE links. Have a look at: **http://elecurls.tripod.com/picaxe.htm.** You will be surprised how much these *cheap* chips can do.

— Ted J. Mieske

Response: I am in the process of learning how to program a PICAXE microcontroller. I find it very limited and frustrating because I have used PICBASIC PRO which has more capability. I don't see the utility of using a $10, 20-pin chip when the job could be done with a $1.50, 16-pin chip. If a reader specifically asks for a PICAXE solution, I may do it, but otherwise I am staying away from the PICAXE.

COUNT DOWN TIMER PARTS LIST				
PART	DESCRIPTION	PKG	PART #	SUPPLY
S1, S2, S3	FOUR POSITION DIP	DIP	696888	Jameco
R1– 24	1/8 OR 1/4 watt, 5%	AXIAL	See Schematic	
R3	50K, 20%, 1/8W, SHORT SHAFT	PANEL	286214	J
C1	10 µF, 16V, 10%, CERAMIC	1206	651495	J
C2	.001 µF, 50V, 10%, CERAMIC 1206	544681		J
C3	.01 µF, 50V, 10%, CERAMIC	1206	87329	J
C4, C5, C6	.1 µF, 50V, 10%, CERAMIC	1206	87388	J
IC1, IC3, IC6	UP/DN COUNTER 40192N	DIP-16	595-CD40192BEE4	Mouser
IC2, IC5, IC7	LCD DRIVER, CD4055N	DIP-16	595-CD4055BE	M
IC4	555 TIMER	DIP-8	27422	J
IC8	OP-AMP, RAIL/RAIL, DUAL	DIP-8	277704	J
IC9	FOUR INPUT AND/NAND			
	EITHER WILL WORK	DIP-14	676123	J
DISPLAY	LCD-S401C71TR	40 PIN	696-LCD-S401C71TR	M
SW1	MOMENTARY P.B.	PANEL	26623	J
SW2	SPST TOGGLE	PANEL	72161	J
SOCKET	20-PIN SIP (TWO NEEDED)		101283	J

J = JAMECO; M = MOUSER

■ FIGURE 6

■ FIGURE 8

```
REM LARGE CLOCK PROGRAM FOR PICAXE 28M 9/9/2009
REM INITIAL CONDITIONS
LET PINS = %00000000                 REM MAKE ALL OUTPUTS LOW

REM PORT B IS NOT A PORT!
SYMBOL H = B1
SYMBOL M = B0

START:

       FOR H = 1 TO 12                          REM HOUR LOOP
              FOR M = 16 TO 112      step 16             REM MINUTE LOOP
              IF PORTA pin0 = 1 then GOSUB SETMIN
                                       'REM BUTTON SWITCH TO SET MINUTES
              IF PORTA pin1 = 1 then GOSUB SETHR
                                       'REM BUTTON SWITCH TO SET HOUR

              IF PORTA pin2 =1 THEN GOSUB LINE3
                                       'REM THIS MAKES A 30 SECOND ADJUSTMENT
                                       'TO INCREMENT THE MIN. LED EARLIER

              LET M=M+H
                     LET PORTC = M          REM INCREMENT THE MINUTE LED
                     PAUSE 60000                   REM PAUSE 1 MINUTE FOR A
                     PAUSE 60000                   REM TOTAL OF 5 MINUTES
                     PAUSE 60000
                     PAUSE 60000
                     PAUSE 60000
              NEXT M

       NEXT H
              GOTO START             REM REPEAT FOR ANOTHER 12 HOURS

SETMIN:
        M=H+M+16
         PORTC = M
        PAUSE 2000                      'REM PAUSE 2 SECONDS
       IF PORTA pin0 = 1 then GOTO SETMIN
                                       'REM HOLD THE M BUTTON DOWN UNTIL
                                       'THE DESIRED MINUTE IS REACHED
       IF PORTA pin2 = 0 THEN RETURN
       ENDIF                           'REM RELEASE THE M BUTTON TO
                                       'STOP INCREMENTING THE MINUTES
       GOTO SETMIN                     'DOESN'T GET HERE IF PORTA pin2 = 0

SETHR:
       H=M+H+1
        PORTC = H
       PAUSE 2000
       IF PORTA pin1 = 1 THEN GOTO SETHR
                                       'REM HOLD THE H BUTTON DOWN UNTIL
                                       'THE DESIRED HOUR IS REACHED
       IF PORTA pin1 = 0 THEN RETURN
                                       'REM RELEASE THE H BUTTON TO STOP
       ENDIF                           'INCREMENTING THE HOUR
              GOTO SETHR               'DOESN'T GET HERE IF PORTA pin3 = 0

LINE3:        PAUSE 30000              REM PAUSE 30 SECONDS

       IF PORTA PIN2 = 1 THEN GOTO LINE3
                                       REM HOLD THE BUTTON DOWN FOR AS
                                       REM MANY 30 SECOND INCREMENTS AS
                                       REM YOU WANT TO CHANGE THE TIME
       IF PORTA PIN2 = 0 THEN RETURN
       ENDIF
END
```

get it all off. Check the pattern for breaks; there usually are a few. I use fingernail polish thinned with acetone to make repairs. An artist's brush might be better, but I use a toothpick to spread the polish. **Figures 3** through **5** demonstrate the process. **Figure 6** is the Parts List.

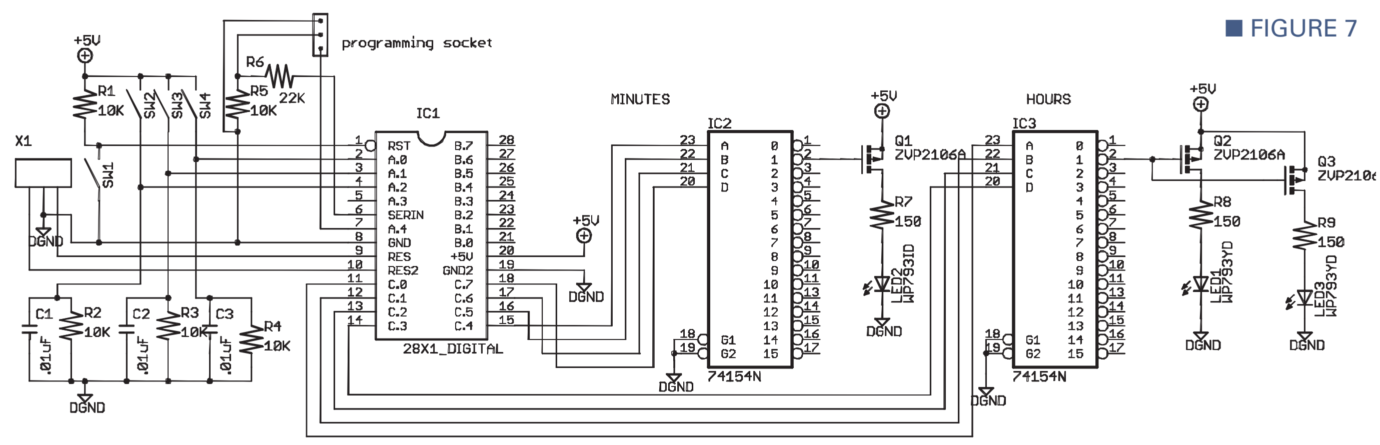
If I were to do this again, I would take more care to make all the solder connections on the back side because removing a part that has been soldered on both top and bottom is difficult; plus, if you want to use sockets, you can't solder on the top side. I recommend using a socket for the display because soldering heat could damage it.

I have not completed building the circuit at this point and the deadline is upon me but by the time this goes to press, I will have the layout fixed and posted. I think I'll also post the Gerber files in case anyone wants to have commercial boards made with plated thru holes. The layouts are available on the *Nuts & Volts* website; look for "Countdown Timer layout."

LARGE CLOCK

Q I need help for a project that I would like to build. I have a large analog clock, (six foot diameter) on an outbuilding. I would like to have 12 red LEDs to follow the hands around in steps of five minutes and 12 green to step the hours off. I would like to use a PICAXE 28 with

■ FIGURE 7



74xx595 output expansion. I would also need some means of setting the minutes and hours.

— Eric Fulton

A I can foresee a problem in that the PICAXE frequency will not be exact so you will have to periodically reset the time. Some feedback could be done to automatically synchronize the clock and PICAXE but I am not going to be that complicated. The 74xx595 is an eight-bit shift register; you would need two of them for 12 LEDs. I will go with a 74xx154 — a four to 16 decoder — because the PICAXE has plenty of outputs to drive them. As I delve into the PICAXE basic, I see that there is only one programmable port. That is a severe limitation but I can figure a way around it.

Only four bits are needed for the hour and minute LEDs (12 = 1100 in binary). The port is eight bits, so I can use the upper four for minutes and the lower four for hours. The first minute is 10000 which is 16 in decimal, and the second minute is 100000 which is 32 in decimal. The difference is 16 and each minute increments by 16.

I think having the minutes LED light 2-½ minutes before the time and stay lit for 2-½ minutes after — so that there is always one on — is the best way to do it. Doing the same for the hours, however, would be confusing so I will have the hours LED light on the hour and stay lit until 59 minutes after. The time setting buttons are used to set this up. My first attempt at a program did not work because I thought I could program the B port as a port, but I can't (dumb system). Each of the B port outputs have to be programmed individually (pain). My comments in the program should be sufficient explanation of how it works. I planned to test it, but I was shipped a nine-pin serial cable instead of the USB that was ordered and it gives an error message "hardware not found." I have ordered a USB cable from SparkFun and will let you know next month if it actually works.

The wire diagram is **Figure 7** and the program is **Figure 8**. In **Figure 7**, I show only one of the 12 LED drivers to save space. The transistor is a P-type MOSFET available from Mouser. For the hour LED driver, I show how you would connect multiple LEDs in case you want to have a cluster instead of a single LED. **NV**

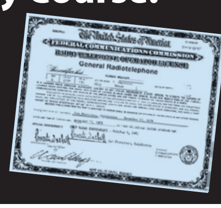
EARN MORE MONEY!

Be an FCC Licensed Wireless Technician!

Make $100,000 a year with NO college degree

Get your "FCC Commercial Radiotelephone License" with our proven Home-Study Course!

- No need to quit your job or go to school.
- This course is easy, fast and low cost.
- No previous experience needed!
- Learn at home in your spare time!



Move to the front of the employment line in Radio-TV, Communications, Avionics, Radar, Maritime and more… even start your own business!

Call now for FREE info:

1-800-932-4268 ext. 209

Or, email us:
fcc@CommandProductions.com

Send to: COMMAND PRODUCTIONS
Warren Weagant's FCC License Training
P.O. BOX 3000, DEPT. 209 • SAUSALITO, CA 94966
Please rush FREE info kit today!

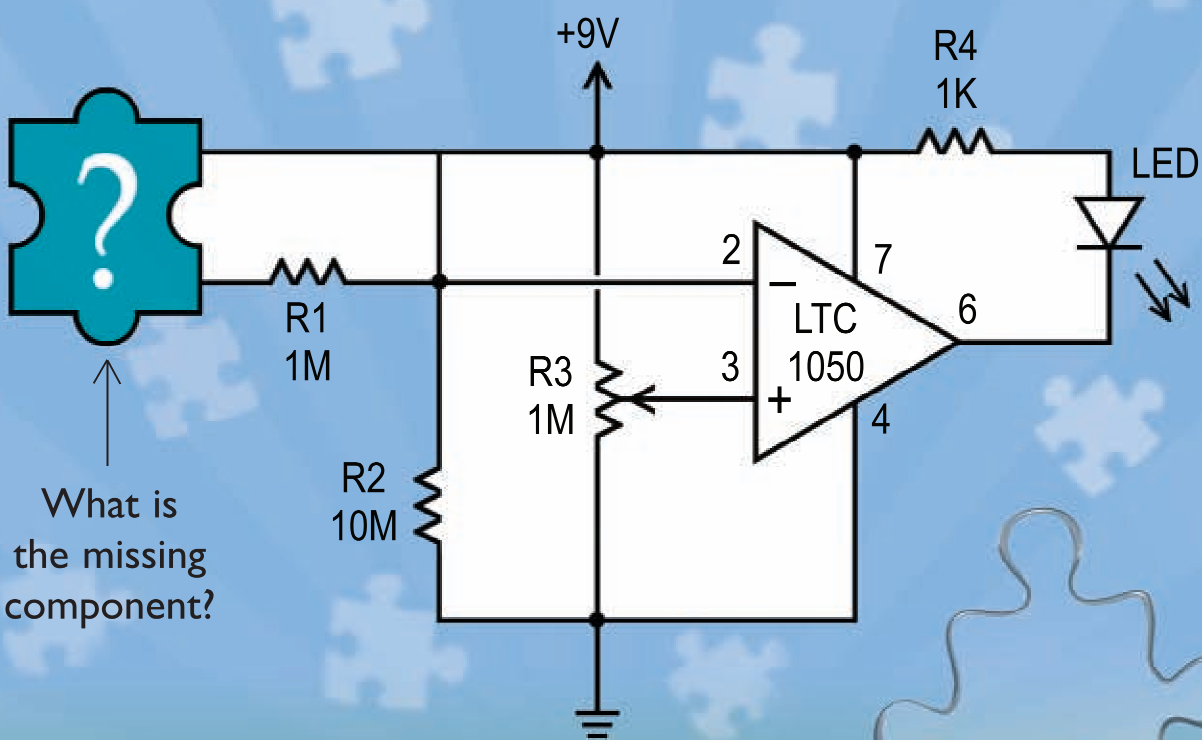
NAME: ____________

ADDRESS: ____________

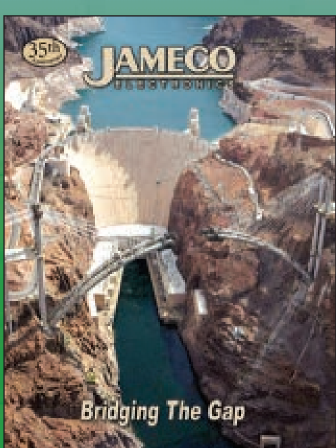
CITY/STATE/ZIP: ____________

You may fax your request to: 415-332-1901

Are you up for a challenge?



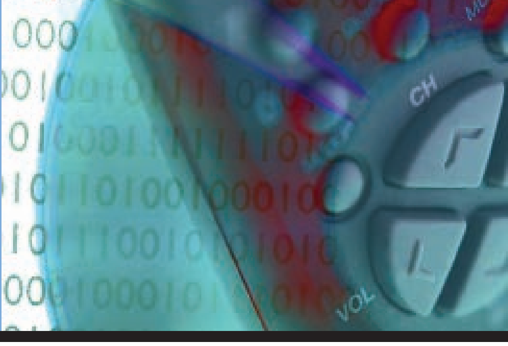
What is the missing component?



Industry guru Forrest M. Mims III has created yet another stumper. The Ultra Simple Sensors Company assigned its engineering staff to design a circuit that would trigger an LED when a few millimeters of water is present in a basement or boat. What is the water sensor behind the puzzle piece? Go to www.Jameco.com/search5 to see if you are correct and while you are there, sign-up for our free full color catalog.

JAMECO® ELECTRONICS

1-800-831-4242 | www.Jameco.com



NEW PRODUCTS

- HARDWARE
- SOFTWARE
- GADGETS
- TOOLS

EXPANSION OF WAVE OSCILLOSCOPE SERIES

LeCroy® Corporation has announced that it is expanding the popular WaveAce™ oscilloscope line to include four channel models from 60 MHz to 300 MHz, and is adding a new two channel, 40 MHz model. The four channel models provide 10 kpts/ch memory and up to 2 GS/s sample rate; the 40 MHz model provides 4 kpts/ch and a sample rate of up to 500 MS/s. All models offer long memory, color displays, extensive measurement capabilities, and advanced triggering to improve troubleshooting and shorten debug time. With USB host and device ports (plus a LAN connection), the WaveAce oscilloscopes easily connect to a memory stick, PC, or printer for saving data or remote control. Combined with the streamlined, time-saving user interface, these features make the WaveAce oscilloscopes the ideal tools for affordable design, debug and troubleshooting from 40 MHz to 300 MHz. The long memory available allows users to capture full sample rate acquisitions that are two to three times longer than the competition. With 32 built-in automated parameters — including advanced timing parameters for skew, phase, and edge-to-edge measurements between channels — the WaveAce oscilloscope broadens the ways a user can understand and analyze waveforms. Additional features such as Pass/Fail testing, user-definable digital filters, and a waveform sequence recorder simplify and shorten debug time.

A new remote control command set has been implemented in the four channel models to make the WaveAce even more useful and versatile. This modern remote control command set provides access to all of the WaveAce controls, functions, and measurements from a remote PC via the rear panel USB or LAN ports. These connections also allow the user to remotely view the waveform display and access a virtual front panel.

Documenting results and saving screenshots, waveforms, and setups are easy with the WaveAce. Internal storage can hold up to 20 waveforms and 20 setups. Mass storage can be done by connecting a USB memory device directly to the front panel of the oscilloscope. The rear panel USB port allows for direct printer connection to quickly generate hard copies of the screen image.

The high performance and large feature set of the WaveAce is controlled by an intuitive user interface with 11 different languages and streamlined front panel. All important controls and menus are accessed from the front panel with a single button press. All position and offsets can be reset by simply pressing the knob; pressing the V/Div knob will switch between fixed and variable gain and pressing the T/div knob will toggle between zoom modes. Buttons on the front panel that open and close menus or switch modes are backlit to let the user know exactly what mode the WaveAce is operating in. The 40 MHz WaveAce 101 price is $695 while prices for the new four channel models range from $1,690 to $2,790.

For more information, contact:
LeCroy
Tel: **1-800-5LeCroy**
Web: **www.lecroy.com**

ARMORED FIBER OPTIC CABLES

A line of highly flexible, ruggedized fiber optic cables for use in critical aerospace, automotive, industrial, medical, and military applications is being introduced by JEM Electronics, Inc., of Franklin, MA.

JEM armored fiber optic patch cords are designed to withstand twisting, kinking, crushing, and prevent rodent damage in applications where traditional fiber optic patch cables may fail during installation and operation.

Featuring stainless steel armored

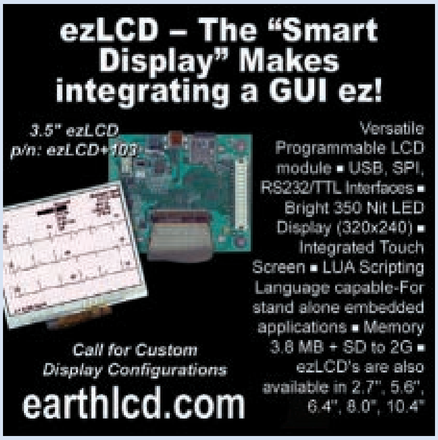
SHOWCASE

flexible tube inside the outer jacket and robust connectors, they are offered in 9/125, 50/125, and 62.5/125 µm sizes and custom lengths for use in a wide range of harsh environments.

Available with SC, ST, MTRJ, FC, MU, E2000, SUS, and other connector types, these fiber optic patch cords come in single- and multi-mode styles up to 500+ meters long. Insertion loss is ≤ 0.2 dB, return loss is □ 55 dB, and repeatability is ≤ 0.2 dB (single mode PC) and the stainless metal tube tensile strength is ≥ 20 Kgf. The firm is ITAR certified, meets ISO 9001-2008, TS16949, and ISO13485 specifications, and are RoHS compliant. The patch cords are priced according to size, length, and cable and connector type. Price quotations

For more information, contact:
JEM Electronics, Inc.
23 National Dr.
Franklin, MA 02038
Tel: **508-520-3105**
Fax: **508-520-3109**
Web:
www.jemelectronics.com

SHOWCASE


ezLCD – The "Smart Display" Makes integrating a GUI ez!
3.5" ezLCD
p/n: ezLCD+103
Versatile Programmable LCD module ▪ USB, SPI, RS232/TTL Interfaces ▪ Bright 350 Nit LED Display (320x240) ▪ Integrated Touch Screen ▪ LUA Scripting Language capable-For stand alone embedded applications ▪ Memory 3.8 MB + SD to 2G ▪ ezLCD's are also available in 2.7", 5.6", 6.4", 8.0", 10.4"
Call for Custom Display Configurations
earthlcd.com


16-bit microcontroller with resident BASIC
* Host on virtually any platform or OS
* 87 I/O pins, 16-channel 10-bit ADC
* No software to install
* USB or RS232
$75 or less*
Easy object-based programming using any terminal program!
* Educational discounts available
www.soBASICsoEasy.com


USB Oscilloscope for $169.50
8+8 digital + 2 analog channels
Oscilloscope - Generator - Logic Analyzer
with UART, SPI, I2C and 1-Wire interfaces
Great for PIC and AVR projects!
www.HobbyLab.us


QKITS.COM
Power Supply: 12/24VDC Load Voltage
30A PWM DC Motor Speed Control
Control the speed of your electric motors or the brightness of your DC light bulbs.
$24.95
Quantity discounts available.
1-888-GO-4-KITS


www.Primecell.com
Battery rebuilding service
Dead Batteries ? Don't toss them.
Send them to us - our rebuilds are better than original specifications.
Tools
Hilti Skil
Milwaukee
Panasonic
B&D DeWalt
Makita All
2-36 Volts
Electronics
Bar Code
Scanners
Surveying
Printers
Laptops
Photography
Radios
APELCO
UNIDEN
G.E. ICOM
KENWOOD
MOTOROLA
MIDLAND
MAXON
YAESU
ALINCO
Visit www.primecell.com for important details
24 Hr Secure recorder tel-fax (814) 623 7000
Quotes email: info@primecell.com
Cunard Assoc. Inc. 9343 US RT 220 Bedford PA 15522


BlinkM
Arduino
I2C RGB "Smart LED"
USB - BT - XBEE
Prototyping Supplies
Serial LCD
FunGizmos.com
RS232, I2C, SPI
$24
Save 5% with coupon code NV
FunGizmos.com

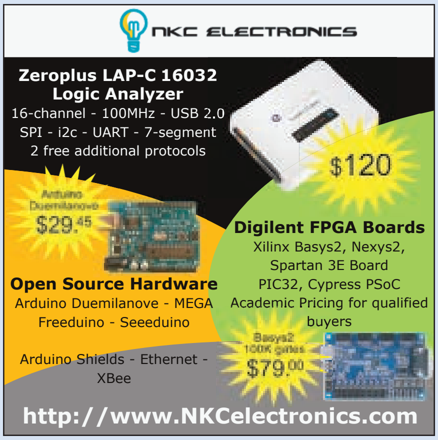

ActiveWire® USB
Simple USB Interface!
Works with MacOS 8/9, Win98/2K/ME/XP FreeBSD and Linux!
24Mhz CPU core with USB
Firmware downloadable via USB
16 bit parallel Input/Output
See web-site for add-on boards
All drivers, manuals, demos are on our web-site for immediate download!
$59
plus shipping
ActiveWire, Inc.
www.activewireinc.com
ph +1.650.465.4000 fax +1.209.391.5060


HYDRA
Game Development Kit
Develop Games, Graphics, and Media Applications with the Propeller™ Powered HYDRA Game Console Kit.
Master multicore techniques & programming while having fun!
www.XGAMESTATION.com


What Your Electronics Store Used To Be
DesignNotes.com, Inc.
Receive a FREE Gift with orders over $100.00!
1-800-957-6867
www.DesignNotes.com


RF Specialists
FCC Part 90 Compliant
manufactured by RADIOMETRIX
USX2
NBFM Multi-channel UHF Transceiver with Programmable RF Power
UHX1
NBFM Multi-channel 500mW VHF Transceiver
NEW!
QPT1
VHF Narrow Band FM 2 Watt Multi-channel Transmitter
NEW
QPX1
2 Watt Multi-channel VHF Transceiver
SMX1
VHF Narrow Band FM Multi-channel Transceiver
RX1M
VHF Narrow Band FM Multi-channel Radio Receiver
LMR1
VHF NBFM Low Cost Multi-channel Radio Receiver
LEMOS INTERNATIONAL
www.lemosint.com
866.345.3667
sales@lemosint.com


Add USB to Your Projects
USB Complete
The Developer's Guide
New Fourth Edition includes USB 3.0 & SuperSpeed!
USB COMPLETE FOURTH EDITION
THE DEVELOPER'S GUIDE
JAN AXELSON
Jan Axelson
Lakeview Research
www.Lvr.com
504 pages
$54.95
From the author of Serial Port Complete


NKC ELECTRONICS
Zeroplus LAP-C 16032 Logic Analyzer
16-channel - 100MHz - USB 2.0
SPI - i2c - UART - 7-segment
2 free additional protocols
$120
Arduino Duemilanove
$29.45
Open Source Hardware
Arduino Duemilanove - MEGA
Freeduino - Seeeduino
Arduino Shields - Ethernet - XBee
Digilent FPGA Boards
Xilinx Basys2, Nexys2, Spartan 3E Board
PIC32, Cypress PSoC
Academic Pricing for qualified buyers
Basys2 100K gates
$79.00
http://www.NKCelectronics.com

Build This Versatile Digital ALARM CLOCK

BY CHRIS SAVAGE

It seems that time is slipping away at an ever increasing pace. This is especially true when we're doing the things we enjoy. Hence the saying, "time flies when you're having fun." When I was younger, I always had a watch so I'd know when it was time to go home, go to bed, etc. As I got a little older and ventured out on my own, I didn't care as much about time and stopped wearing a watch altogether (the late nights and early mornings seemed to blend in together). These days, time is more important than ever with a family, job, appointments, and meetings. I find that I don't have *enough* devices that tell time (watch, computer, iPod, etc.). Suddenly, I developed a fascination with timekeeping and found myself compelled to build my own clocks and timers. The real bonus here is that once you've mastered keeping time, you can do all kinds of things with it — from the way you display it, to the way you signal specific points in time (alarms).

The Heartbeat of Time

In order to keep track of time, we'll need a device that can accurately update itself, even when power is lost. For me, that device is the DS1302 from Maxim Integrated Products. The DS1302 is a battery-backed, real time clock with RAM. The DS1302 counts seconds, minutes, hours, date, month, day, and year with leap year compensation until 2100. It also contains 31 bytes of battery-backed, general-purpose RAM. With a synchronous serial interface and wide operating voltage, the DS1302 is an ideal choice for both 5V and 3.3V microcontrollers.

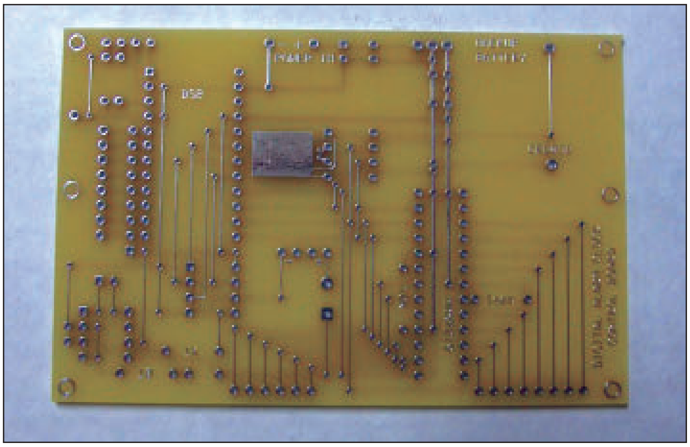
Keeping It Simple

Ever since I built my first clock, I have been inspired to create some unique time-pieces. In the spirit of keeping it simple and to make it easier to learn how you can build your own clock, I will demonstrate a basic design that we have been using in our house for a few years now. The three clocks you see in **Figure 1A**, **1B**, and **1C** are all based on the same control board design. Each is unique in its appearance and also in its function to some degree. By combining the common elements on the control board (such as the clock chip, alarm amplifier, and display driver), we can keep the base programming code simple and common to all three versions while having the ability to customize each clock for basic things such as display and input configuration.

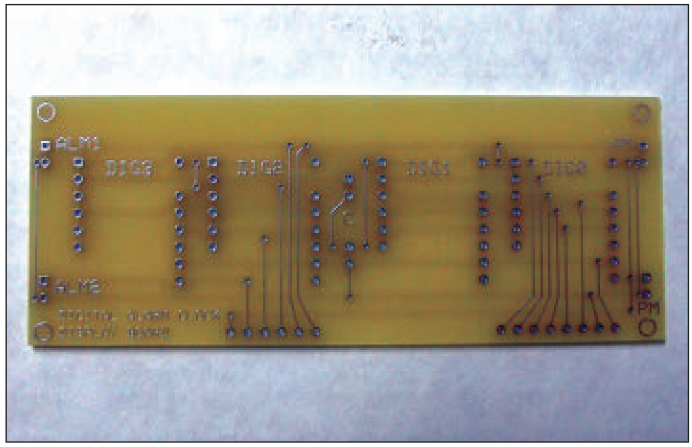
The Control Board

Figure 2 shows the main control board. This board contains a BASIC Stamp 2 microcontroller, a MAX7219

■ FIGURES 1A, 1B, and 1C. Three prototype alarm clocks.



■ FIGURE 2 -The main control board.



■ FIGURE 3 - Seven-segment display board.

display driver, a DS1302 RTC, a DS1804 digital potentiometer, an LM386 amplifier, and all the support components. What it doesn't contain is the display or input functions. Instead, an interface is available for both the display (along the bottom of the board) and the input (along the left side). This allows for further customization of the display and input without having to redesign the control board. The eight available I/O all contain 10K pull-up resistors making it easy to connect switches, photocells, thermistors, or even a quadrature encoder for input.

The Display Board

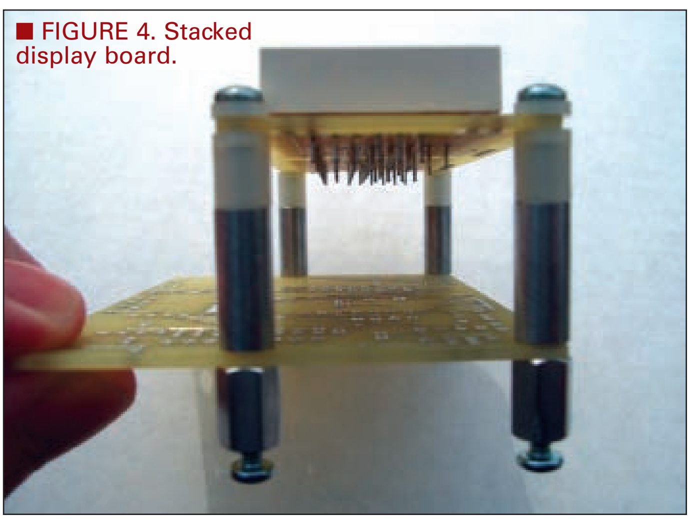
Figure 3 shows the display board. This board was made to connect directly to the control board either by connectors or wires. The mounting holes are aligned with holes on the control board allowing you to stack it right on the board (see **Figure 4**). Trimmed component leads were recycled to connect the display board to the control board (see **Figure 5**). This concept was used for all three clocks and expanded on by connecting the stacked boards directly to each unique enclosure (see **Figure 6**). In populating the display board, I used common seven-segment displays that I got from both Jameco, as well as from some older Parallax kits. These seem to have a common pinout and were available in green or red.

When selecting the T1 LEDs to use for the colon, alarm, and AM/PM indicators, try to use an LED with a similar forward voltage to the seven-segment displays. This will help ensure that the brightness is consistent, especially when using dimming as the MAX7219 maintains current for the entire display. Since the LEDs are free-standing on the board, I used heat shrink tubing to block the light from the side of the displays. Colored translucent plastic was obtained for a filter for both the red and green displays. Be careful when cutting this stuff. I actually cracked a few pieces and had to get more to finish the project. There are guides available on cutting and drilling Plexiglas, so I won't go into that here.

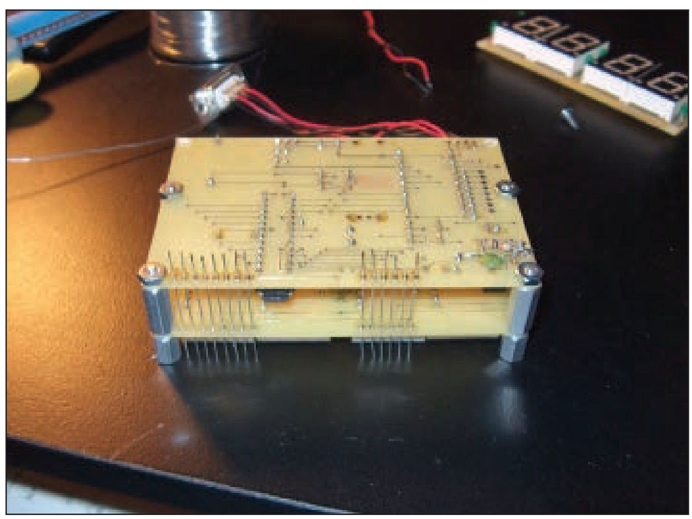
Circuit Design

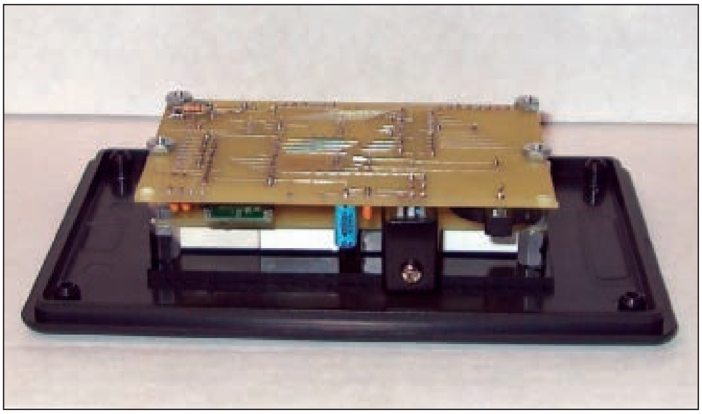
Looking at the schematic (**Figure 8**), you can see the individual blocks for each project all broken down in their basic components. What the schematic does not contain

■ FIGURE 4. Stacked display board.

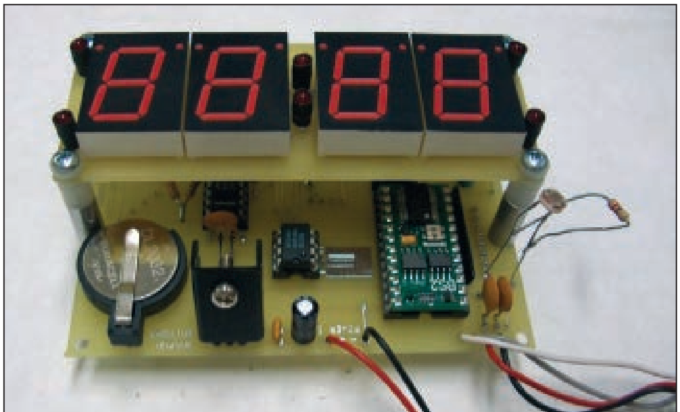


■ FIGURE 5. Component leads used to connect the display board to the control board.





■ FIGURE 6. Display board mounted to enclosure.



■ FIGURE 7. Display board details.

are the inputs for setting time, etc. These are customizable and are slightly different on each clock. So, we'll cover the design of the common components.

The BASIC Stamp 2 is the brains of the operation here. I could have used a more powerful BASIC Stamp model and probably would have if I had used a rotary encoder on any of these, but I decided to keep it simple. The Stamp drives the display using the MAX7219 driver IC. While a little pricey, I really like this chip and find that it greatly simplifies things, especially if your display contains more than four digits (adding seconds). The DS1302 is on the same synchronous serial bus as the MAX7219 which saves two I/O lines. The DS1804 is used in this design to provide a volume control for the LM386 audio amp IC. The battery backup keeps the time when the power fails.

Design Pitfalls

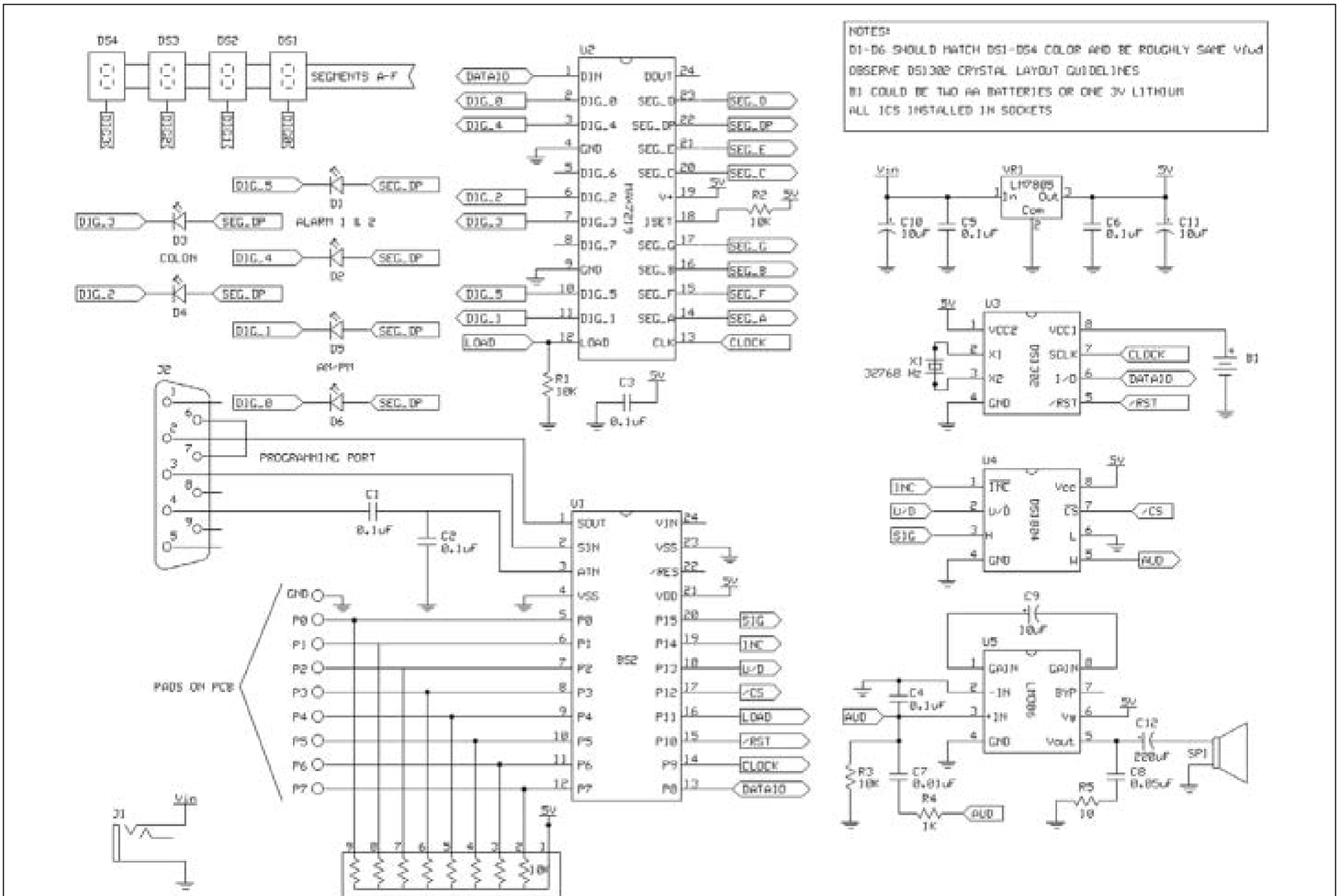
As with any project, when you change something even a little bit from what you started with, your outcome can vary significantly. This was the case during the original build of these clocks. You may have noticed in **Figure 5** the extra components tack-soldered on to the back of the control board. When I prototyped the design, I used the Parallax Professional Development Board — my favorite resource for Stamp prototyping. I used the on-board amplifier for testing sound output, but when I designed the LM386 amplifier in the circuit I first built, I used a generic configuration. Imagine my surprise when the alarm sounded not only different, but had poor volume control and range. The patches were made to attempt to correct this and I have updated the ExpressPCB files (available in the download package at **www.nutsvolts.com**) accordingly (so you don't have to do this if you have a board made).

Clock Features

At first, these would seem to be ordinary digital clocks. However, each clock has additional features. Some are common to all three clocks such as

BILL OF MATERIALS

QTY	DESIGNATOR	COMPONENT	SOURCE	PART #
1	PCB1	Digital Alarm Clock Control PCB	ExpressPCB	N/A
1	PCB2	Digital Alarm Clock Display PCB	ExpressPCB	N/A
1	U1	BASIC Stamp 2 Module	Parallax Inc.	BS2-IC
1	U2	MAX7219 8-Digit 7-Segment Driver	Digi-Key	MAX7219CNG+-ND
1	U3	DS1302 Real Time Clock w/RAM	Parallax Inc.	604-00005
1	U4	DS1804 Digital Potentiometer	Digi-Key	DS1804-100+-ND
1	U5	LM386 Audio Amplifier IC	Digi-Key	LM386N-3-ND
1	VR1	LM7805 5V Voltage Regulator	RadioShack	276-1770
6	D1-D6	Red T1 LED	RadioShack	276-026
4	DS1-DS4	Green 7-Segment LED Display	N/A	* See Article Text
1	RN1	10K, 8 Bussed	Jameco	857080
3	R1-R3	10K, 1/4W, 5%	Parallax Inc.	150-01030
1	R4	1K, 1/4W, 5%	Parallax Inc.	150-01020
1	R5	10 ohm, Metal Oxide	Parallax Inc.	150-01000
6	C1-C6	0.1 µF, 50V, 20%, Mono, Radial	Parallax Inc.	200-01040
1	C7	0.01 µF, 50V, 20%, Poly, Radial	Parallax Inc.	200-01031
1	C8	0.05 µF, Ceramic, Radial	N/A	* See Article Text
3	C9-C11	10 µF, 35V, Electrolytic, Radial	Parallax Inc.	201-01060
1	C12	220 µF, 35V, Electrolytic, Radial	RadioShack	272-1029
1	X1	32768 Hz Crystal, 6 pF	Parallax Inc.	251-03230
1	J1	2.1 mm Barrel Jack	Digi-Key	CP-202A-ND
1	J2	DB-9 Connector - Female	RadioShack	276-1538
1	B1	3V Lithium Battery	N/A	* See Article Text
1	SP1	Speaker	N/A	* See Article Text
0	ENC1	Project Enclosure (Optional)	N/A	* See Article Text
0	ENC2	Project Enclosure (Optional)	N/A	* See Article Text
0	ENC3	Project Enclosure (Optional)	N/A	* See Article Text



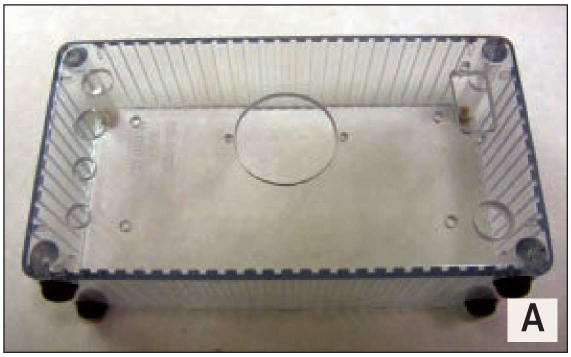
■ FIGURE 8. Overall schematic.

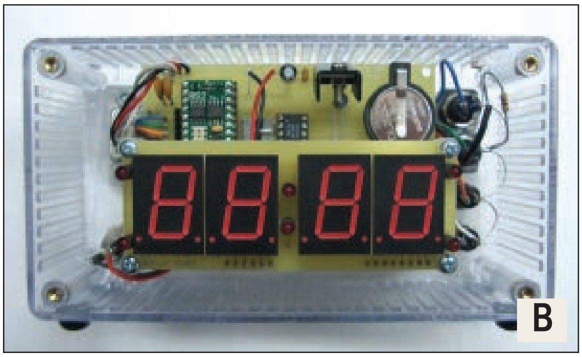
crescendo alarm capability, multiple alarm times, 12 or 24 hour format, battery backup, and dimmable display. Other features are unique for a particular clock or are modified to act differently than the others. For example, the clock with the translucent case uses a photo resistor to help the clock determine if it is light or dark in the room. This data is used to adjust the brightness of the display. Another clock has a programmed set of times in which it brightens or dims its display. These settings are stored in the battery-backed RAM and restored on power-up. The crescendo alarm capability allows the clock to start the alarm at the lowest volume and ramp it up as time goes on at a rate you select. In fact, you can even change the alarm sound itself. There are many customizable options here and very little code needs to be modified to enable these features. The base code contains subroutines that handle all the details such as reading/setting the time, reading/writing to the RAM, etc.

Parts Resources

What? You don't have that in your junk box? No problem! I listed some sources for the components. Everything used was something I had in my spare parts cabinet. Two of the enclosures came from RadioShack. The translucent enclosure was purchased from a surplus dealer such as All Electronics, although it has been a few years so you might need to look around. I remember

■ FIGURES 9A and 9B. Case milling and installation.





those enclosures were available in red, clear, and blue, so I took the red cover from one and the clear case from another to make the clock more techie looking. During prototyping, I used two AA batteries for the battery backup, but then I found some CR2032 coin cell battery holders on eBay and bought a dozen or so. That's what I used, so if you have the board made from ExpressPCB, remember you may need to make some changes if your battery holder mounts differently or if you use a different type of backup battery altogether. Please feel free to adjust the components to what you have on hand. It's a good way to make use of those spare parts.

Customizing Things

In fact, you should always feel free to experiment with the elements of a design. It can be just a matter of aesthetics or perhaps that the speaker isn't loud enough for you. Just be sure you've tested those changes before you commit to a PCB (lest you end up having to patch your design like I did). Of course, these three clocks have been running for over three years now and there is very little I would change in the overall design if I started over.

In a future article, I will show you a very interesting variation on the concepts here. A hybrid binary/digital clock which displays the time in the traditional numeric form we're all used to (12 and 24 hour modes supported), as well as in binary with each one having a matching binary representation of itself displayed in real time. The day of the week, the date, and temperature are also available on a separate LCD display.

Everything talked about here can be found on the *Nuts & Volts* website. Additional information, updates, and forum discussions can be found by visiting the project website located in the resources sidebar. This is a great place to see what others have done and learn how to make specific modifications to your clock. Take care and have fun! **NV**

RESOURCES

Project Page
www.savagecircuits.com/alarmclock/

Project Discussion
http://forums.savagecircuits.com/index.php?topic=35.0

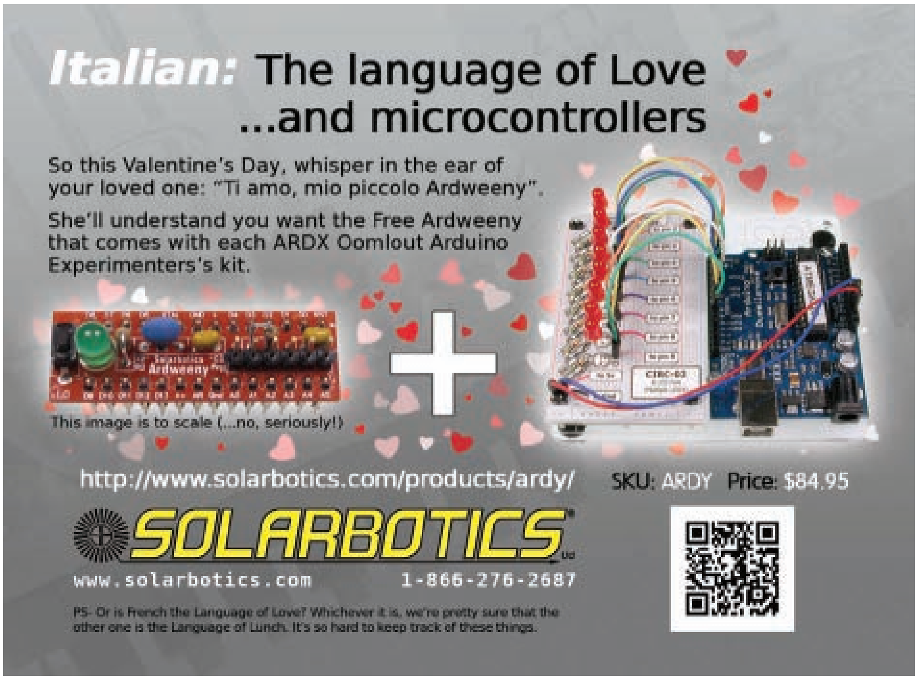
Parts Resources:

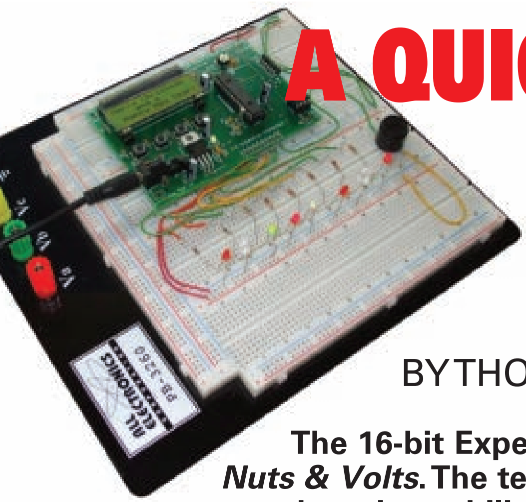
Parallax, Inc.
www.parallax.com

Digi-Key
www.digikey.com

Jameco
www.jameco.com

RadioShack
www.radioshack.com





A QUICK TOUR OF THE 16-BIT MICRO EXPERIMENTER MODULE

BY THOMAS KIBALO

The 16-bit Experimenter Module was introduced in the December '09 issue of *Nuts & Volts*. The technology is exciting and we are going to discuss several of its on-board capabilities this month. We'll supplement some of this discussion with experiments to allow you to gain a better understanding of how you can use the Experimenter for your own applications. All the experiments are written in C code using our own custom function library. We'll try to keep things as straightforward and simple as possible, but you may need to brush up on some of the basic C syntax.

Let's kick it off with a little more in-depth look into the microcontroller itself before going into the Experimenter's LCD display, buttons, EEPROM, I/O Expansion Bus (see the **block diagram**), and the supporting software library.

The Microcontroller, ICSP, and Required Tools

We talked about the PIC24FJ64GA002, the core processor on the Experimenter, in the last article, but let's take a closer look. First of all, as Microchip's lowest cost 16 bit microcontroller, it executes up to 16 Million Instruction per second (MIPS) rate. The internal peripheral content is impressive: 21 programmable I/O; 500 Ksps ADC with up to 10 analog channels; two analog comparators; five 16-bit timers; five input capture and five output Compare/PWM modules; dual UART; and dual SPI and I^2C. A real benefit with the 16-bit Experimenter is that you can access all of these peripherals for your solderless breadboards. The Experimenter's In Circuit Serial Programmer (ICSP) is based on the proprietary Microchip six-pin serial interfaces. It is incorporated as an external connector. The ICSP allows the PIC24FJ64GA002 64K Flash memory to be reprogrammed many times. For tools, you need to get the Microchip Free MPLAB, their integrated development environment (or IDE), and their limited free student edition of the P24 C Compiler, as well as the PICKIT2.

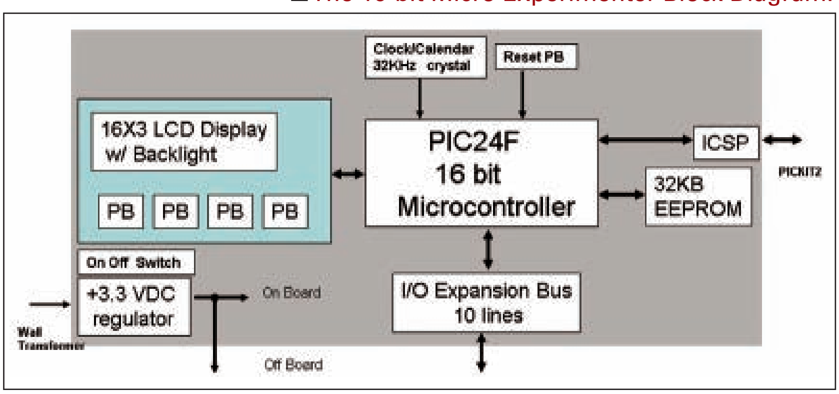
The LCD Display and Button Hardware

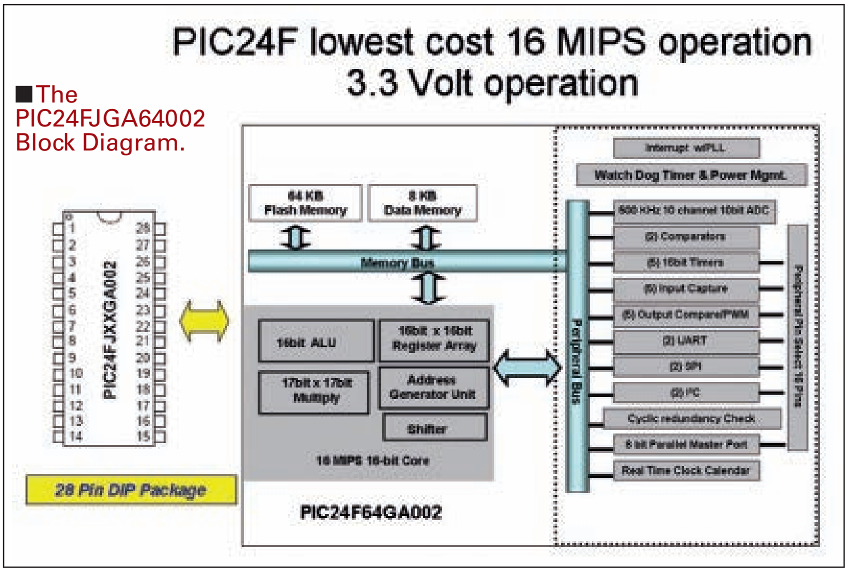
The display uses a +3.3 VDC 16x3 character LCD display offered by Electronic Assembly. It is very similar in control to other LCDs with an extended feature of built-in contrast control (no need for external potentiometers) and very low power. Designed for compact, hand-held devices, it is ultra-flat. The display interfaces to the PIC24F using four-bit data and a two-bit command digital interface. Each of four data lines connecting the PIC24F with the LCD are shared with one of the pushbuttons. The PIC24F services the LCD display as needed, and when the display is not being used the PIC24F polls the user pushbuttons for activity. The pushbuttons are configured with resistor pull-ups and pull-downs so as not to interfere with LCD operations (even when a button is pressed during a PIC24F write to the LCD).

Driving the Display — LCD Demo

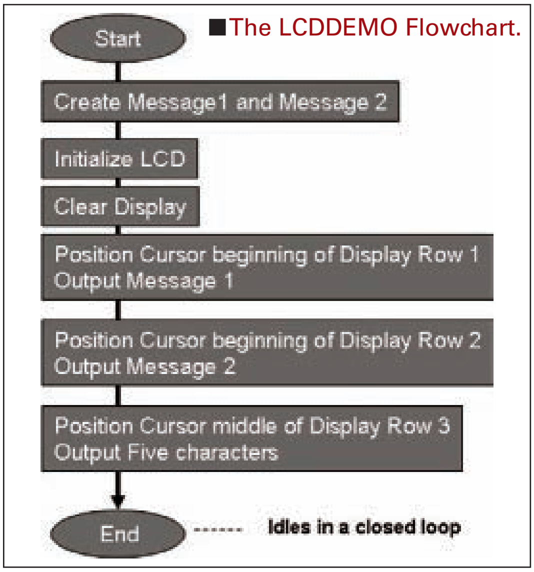
So much for hardware, now here's the fun part. Let's learn some software functions for driving the LCD and how to use them. Here are the five we will be working

■The 16-bit Micro Experimenter Block Diagram.





■The PIC24FJGA64002 Block Diagram.



■The LCDDEMO Flowchart.

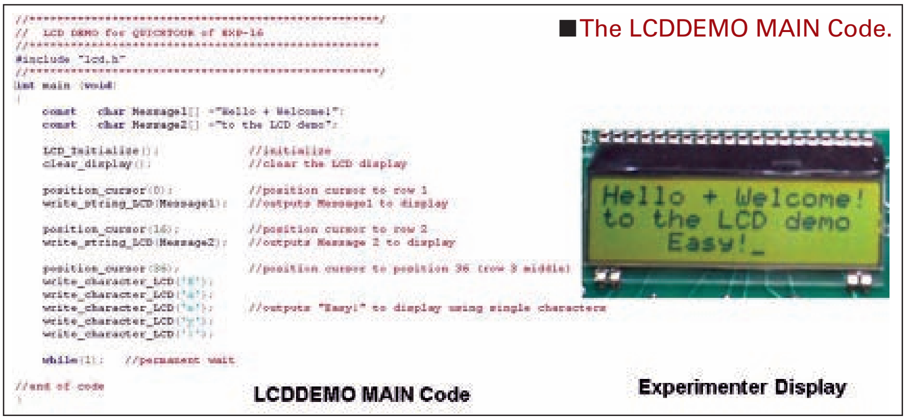
with and a description of what they do. Their titles pretty much explain what they do.

- **LCD_Initialize ()** — Initializes the LCD display and makes it ready for output. This needs to be called before any other LCD function is called.
- **clear_display ()** — Clears the display and places the cursor at row 1 column 1 (top left corner).
- **position_cursor (position)** — Places the cursor at a designated position; a number from 0-47.
- **write_string_LCD (string name)** — Writes a string to the display at the current cursor position (see **explanation later**).
- **write_character_LCD (character)** — Writes a single character to the display at the current cursor position.

An example project LCDDEMO is supplied. This project shows how to use each of the above LCD functions and how to call them from your MAIN file. Let's walk through the flowchart and then the code.

LCDDEMO starts by declaring two strings: message #1 and message #2. Each message is configured to be under 16 characters to keep the display nice and neat; that is, keeping each message to within a single display row. The next step is to initialize the LCD and clear the display contents before use. Then, we position the cursor to the beginning of row one (cursor position = 0) and write out message #1. Next, we position the cursor at the beginning of row two (cursor position = 16) and write out message #2. Just to keep things interesting, we'll position the cursor in the middle of line three (cursor position = 36) and write out five characters. If you haven't figured it yet, cursor positions 0 to 15 correspond to the top row (row #1) of the LCD display. Positions 16-31 are row #2 and positions 32-47 correspond to row #3. The LCD is initialized so that the cursor itself is not visible; the cursor position automatically advances to the next position when data is written to the display. Any data that is written must be in ASCII format.

A copy of the LCDDEMO MAIN code is shown **in the figure** along with the resulting Experimenter LCD display, from when the code is executed. Notice that we have #include "lcd.h" at the top of the file to allow us to reference the LCD library. Try to match up the code to the flowchart with the code to understand the library syntax. To help you along, there are lots of comments included in the code.



■The LCDDEMO MAIN Code.

Writing to the Experimenter LCD using these library functions is straightforward. To prove this to yourself, try modifying the code to configure your own display message. You'll will find that writing your own messages to the LCD is a snap!

Using the Pushbuttons with the LCD — Button Demo

Pushbuttons are a necessary way to communicate with your Experiment while

applications are running. Let's review some of the software functions for handling the pushbuttons and how to use them. Here are the two functions we will be working with and a description of what they do:

- **BtnInit () —** Initializes the Experimenter pushbuttons SW1 to SW4 for operation. This needs to be invoked before any other button process.
- **process_buttons() —** Returns a value that indicates which Experimenter pushbutton was depressed:
    - If SW1, then returned value = 1
    - If SW2, then returned value = 2
    - If SW3, then returned value = 3
    - If SW4, then returned value = 4
    - Returned value = 0 if no button is pressed.

An example project BUTTONDEMO is supplied that you can use as a template. This project shows how to use each of the above button functions and then how to call them from your MAIN file. In order to make the demo more meaningful, we'll also include the LCD library discussed earlier. Let's walk through the whole thing starting with the flowchart.

BUTTONDEMO MAIN declares four unique messages that will be displayed, depending on which button is pressed. We initialize the LCD, clear the display, and then initialize button operation. You typically do all your initialization up front to get it out of the way. BUTTONDEMO enters a continuous loop that does button processing, determines which of the four buttons was pressed, and then does the specific processing for that pressed button. In our case, we'll simply display a message on the LCD that the button was pressed. It is important to note that the message gets displayed *after* the user presses the button and releases it (that's just the way I wrote the button process function). Keep in mind you need to process all the buttons periodically so they respond to the user in a timely manner. This is why BUTTONDEMO is a continuous loop.

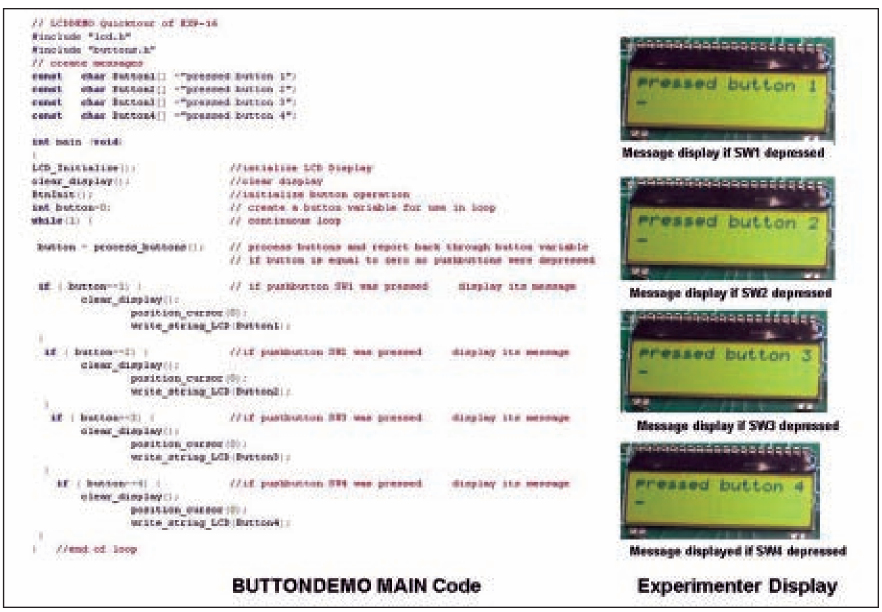
A copy of the BUTTONDEMO MAIN code is shown **in the figure** along with the resulting Experimenter LCD displays for each pushbutton operation. We have #include "lcd.h" and #include "button.h" at the top of the file to allow us to reference both the LCD library and the BUTTON libraries. Again, try to follow the code and compare it against the flowchart. Throughout all these demos, we will incrementally build up the capabilities for the Experimenter by adding and using one library function after another.

The basic structure of BUTTONDEMO makes a good template for applications that may need pushbutton operation. You can simply add the button specific processing for what you need for your own application (The challenge is going beyond simply outputting a message).

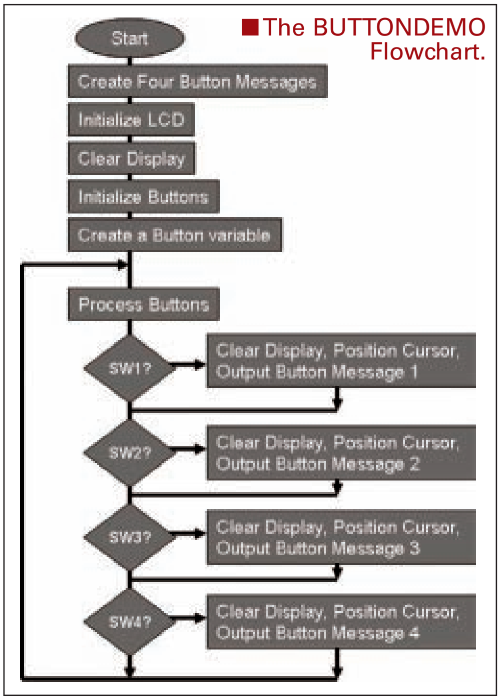
EEPROM Hardware

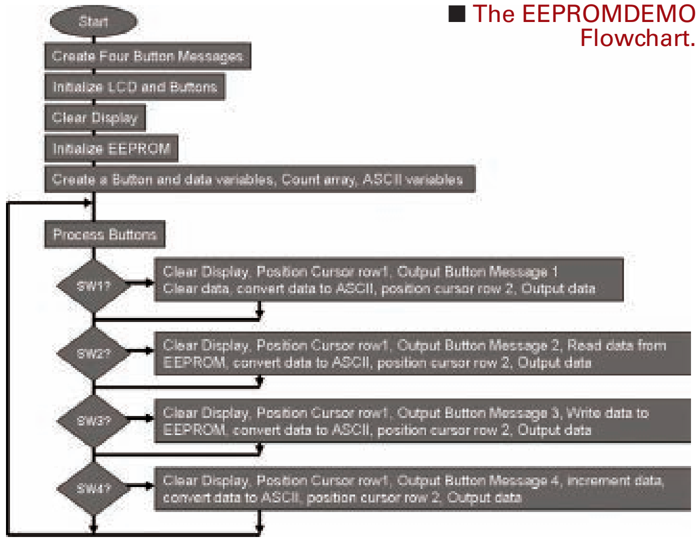
The Experimenter has a 25LC256 EEPROM device that contains 256 Kbits or 32 Kbytes of nonvolatile memory. The 25LC256 is a memory technology that allows us to store data under program control and retain the contents even when power is turned off. This can be very useful for applications requiring password settings or storing data measurements (as in a logging function). The 25LC256 uses a serial communication to talk to the Experimenter's PIC24F. This data interface is called SPI (Serial Peripheral Interface). In our library, SPI communication is handled automatically. Each 25LC256 storage word is two bytes. Given the byte size of the 25LC256, this means we can store up to 16K

■The BUTTONDEMO MAIN Code.



■The BUTTONDEMO Flowchart.





■ The EEPROMDEMO Flowchart.

total. In general, for your experiments, you need to establish what data you want to store and how that data will be organized over the 16K. In a future article, we will actually use this same library capability to configure the Experimenter as a data-logging device for temperature.

Working with the EEPROM, LCD, Buttons — EEPROMDEMO

The EEPROM library is really straightforward. Take a quick look at the functions listed below:

- **InitNVM ()** — Initializes the Experimenter EEPROM for access. This must be called first before any other EEPROM library function is called. NVM stands for Non-Volatile Memory.
- **iReadNVM (address)** — Reads the EEPROM at an address and returns the 16-bit value stored at that location.
- **iWriteNVM (address, data)** — Writes a 16-bit data value to the EEPROM at the designated address.

Using the EEPROM library function, we do the basic initialization of SPI to access the EEPROM, and then read and write from it once it is initialized.

An example project EEPROMDEMO is available for your use. Let's walk through it. A flowchart is provided. To keep things interesting and more meaningful, we will use both the pushbutton and LCD library functions we learned earlier. The demo counts the number of user pushbutton sequences, and then stores this count in EEPROM; it allows for recovery of this count (even when cycling power). All pushbutton actions are at the discretion of the user. The intent of the demo is to allow the user to independently store and update the data (as well as zero it), then cycle power and examine the result. To make things easy, only one address is used (0x1238) but you can try other even address locations. Remember with this eight-bit EEPROM, we need to use even addresses for storage and retrieval of 16 bits.) The result should demonstrate that data contents — as set by the user — can be stored and retrieved using the 25LC256 EEPROM and its library.

You need to make sure that the library is included as files in your own project builds and that "#include NVM.H" is part of your main file. For this demo, we added "BUTTON.H" and "LCD.H" to use their specific libraries, as well. We've actually added several new functions in this demo that have not been covered yet but are worthy of our attention

- **write_array_LCD (array name, number of elements)** — Writes the content of an array for a prescribed number of elements to the LCD display, starting at the current cursor position. This is a method for dynamically updating the display with real data versus fixed strings. This is a member of the LCD function library we hadn't covered yet.
- **binary_to_ASCIIConvert (data)** — This function converts binary data to its ASCII equivalent and places the ACSII values in their proper position in an array. The array is typically used as output to the LCD display using the function described above. The processed data is stored in the following memory locations, as well as inside the array:

  - Bcd10000
  - Bcd1000
  - Bcd100
  - Bcd10
  - Bcdunits

The initialization code for EEPROMDEMO is shown **in the figure**. There are the normal string identification messages, Message Button [] (1 to 4), the data variable itself, and several new twists — a blank array designated count [], variables bcd10000, bcd1000, bcd100, bcdtens, and bcdunits. In addition, there is a new prototype function definition binary_to_ASCIIConvert (). Let's try to put it together.

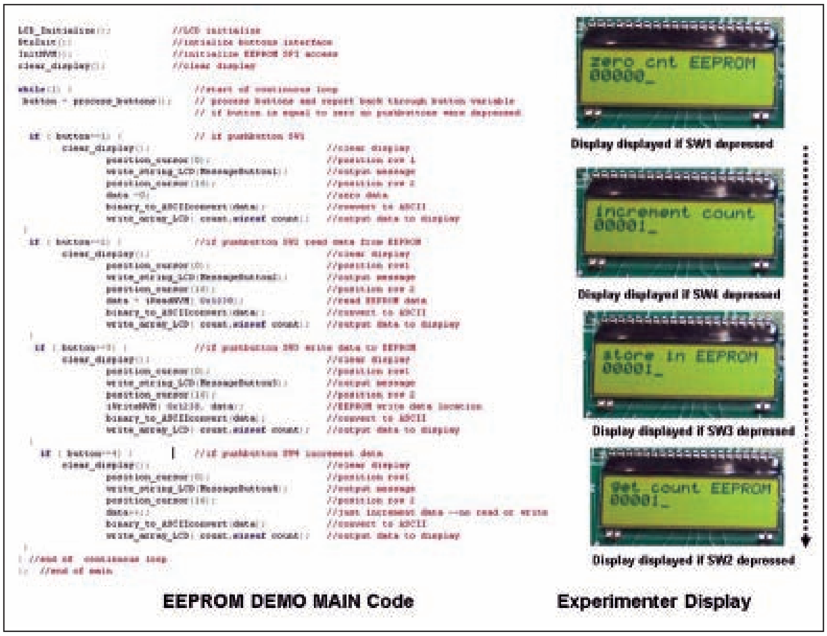
To date, we have not actually presented live data to the LCD display — only fixed strings. The use of count [], bcd10000, bcd1000, bcd100, bcdtens and bcdunits variables, and binary_to_ASCIIConvert () remedy this. The binary_to_ASCIIConvert () takes data as an argument and then processes it to develop the ASCII equivalent in the

```
#include "lcd.h"            // lcd library
#include "buttons.h"        //button library
#include "NVM.h"            //EEPROM library

void binary_to_ASCIIconvert( int n);

const   char MessageButton1[] ="zero cnt EEPROM";
const   char MessageButton2[] ="get count EEPROM";
const   char MessageButton3[] ="store in EEPROM";
const   char MessageButton4[] ="increment count";

char bcd10000 =0;
char bcd1000 =0;
char bcd100 =0;
char bcdtens =0;
char bcdunits =0;
char count[] = {' ',' ',' ',' ',' '};  //modified to display count
int data =0;
int button=0;                  // create a button variable for use in loop
```

■ The EEPROMDEMO Code.

bcd10000, bcd1000, bcd100, bcdtens and bcdunits variables, as well as in its proper position (most significant to least significant digits) in the blank array convert []. This is not the whole story, though. We have another LCD library function "write_array_LCD ()."

This function allows a character array like convert [] to be presented on the LCD display like the string arguments earlier. By updating the convert [] with the ASCII results from binary_to_ASCIIConvert (), we can use write_array_LCD () to update the LCD with the latest data values live. To gain a better understanding, examine the EEPROMDEMO code shown **in the figure** and some of the Experimenter displays that occur as part of your EEPROMDEMO operation. This program structure of updating the LCD display with live data can be an important part in future Experimenter applications.



■ EEPROMDEMO MAIN Code.

Analog Anyone? The I/O Expansion Bus and ADC

The I/O expansion bus is the way in which you can hook up and integrate the Experimenter to other electronics on your solderless breadboard. It will become a major player in future articles. The bus consists of 10 I/O pins that are completely software configurable. As a rule, we typically "program" our pin I/O functions as needed in the beginning of the program and then "Lock" the program for application use. We'll go through several examples over the course of time to help you get a handle on this programmable feature.

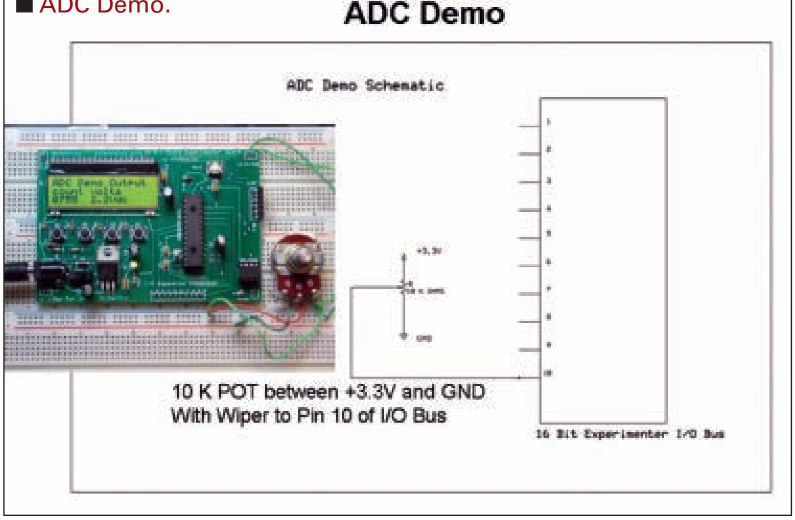
The PIC24 Analog-to-Digital Converter (ADC) is a 10-bit converter that has up to 11 input channels and performs conversions up to 500K per second. With the Experimenter, five of these 11 channels are available as ADC inputs with the I/O expansion bus. They are I/O expansion pins 1, 7, 8, 9, and 10. Using our ADC library, we are able to make any of these pins act as an ADC input and perform an ADC conversion. The conversion automatically converts voltage signals at the input of the pin between 0 volts to +3.3 volts, to a numeric 10-bit value 0 to 1023, representing that voltage. This capability allows the PIC24F to measure external voltage sources (i.e., sensor outputs). Let's try it.

An ADCDEMO program using the ADC library is available. Here we use the library to configure an I/O expansion pin (in this case, pin 10) as an ADC input and then perform a conversion. The conversion is displayed on the LCD in both count and voltage. An external potentiometer (pot) is required to act as a variable voltage source. Hook the wiper of the pot to pin 10 and the other sides of the pot to +3.3V and GND. Since the program has a continuous loop, the conversion and LCD updates are continuously happening. Turn the pot and watch the results. This program uses "#include ADC.H" and #include LCD.H" to reference needed libraries.

The ADC library functions that are used are:

- **InitADC (IOpin) —** Initializes the designated expansion bus pin for ADC operation as an input channel. There are only five possible pins available for use on the I/O expansion. They are designed in the code as: pin1, pin7, pin8, pin9, pin10 (make sure to use the exact syntax as shown here). This function must be called first before any ADC library function. Only one pin can be designated as an input at a time.
- **ReadADC (IOpin) —** This function performs an ADC conversion on the input voltage present on the pin and

■ ADC Demo.



returns an integer value from 0 to 1023 representing the measured voltage.

Examine the MAIN code using MPLAB IDE; it is well documented. Try changing the pin value to one of the other pins, then rebuild and reload the program. Move the pot wiper over to this pin to get readings. The ADC library is an important tool for performing external voltage measurements with the Experimenter. We will use it again in subsequent articles.

Ideas for the Future

We have covered several important subsystems for the Experimenter showing a number of functional capabilities. Coverage included hardware as well as software examples for you to work on. Working the software examples will help to solidify these capabilities for your consideration in your projects. Within each software project, text files are included that fully describe the libraries being used. We are just touching the surface here. There are more features and interesting applications to come. **NV**

The Standard for checking Capacitors in-circuit

Good enough to be the choice of Panasonic, Pioneer, NBC, ABC, Ford, JVC, NASA and thousands of independent service technicians.

Inexpensive enough to pay for itself in just one day's repairs. At $229, it's affordable.

And with a 60 day trial period, satisfaction guaranteed or money-back policy, the only thing you can lose is all the time you're currently spending on trying to repair all those dogs you've given up on.

CapAnalyzer 88A

Locate shorted or leaky components or conditions to the exact spot *in-circuit*

Still cutting up the pcb, and unsoldering every part trying to guess at where the short is?

$229

Your DVM shows the same shorted reading all along the pcb trace. LeakSeeker 82B has the resolution to find the defective component. Touch pads along the trace, and LeakSeeker beeps highest in pitch at the defect's pad. Now you can locate a shorted part only a quarter of an inch away from a good part. Short can be from 0 to 150 ohms.

LeakSeeker 82B

Available at your distributor, or call 561-487-6103

Electronic Design Specialists www.eds-inc.com

It writes your Full Speed USB Code!

NEW! **HIDmaker FS for Full Speed FLASH PIC18F4550**

Creates complete PC and Peripheral programs that talk to each other over USB. Ready to compile and run!

- *Large data Reports*
- *64,000 bytes/sec per Interface*
- *Easily creates devices with multiple Interfaces, even multiple Identities!*
- *Automatically does MULTITASKING*
- *Makes standard or special USB HID devices*

NEW! "Developers Guide for USB HID Peripherals" shows you how to make devices for special requirements.

Both PC and Peripheral programs *understand your data items* (even odd sized ones), and give you convenient variables to handle them.

PIC18F Compilers: PICBASIC Pro, MPASM, C18, Hi-Tech C.

PIC16C Compilers: PICBASIC Pro, MPASM, Hi-Tech C, CCS C.

PC Compilers: Delphi, C++ Builder, Visual Basic 6.

HIDmaker FS Combo: Only $599.95

Trace SYSTEMS, Inc.

DOWNLOAD the HIDmaker FS Test Drive today!
www.TraceSystemsInc.com
301-262-0300

TouchTone Phone CONTROLLER

Build this controller circuit to operate multiple AC devices by phone via wireless transmitters

By JF Mastromoro

In the November '09 issue, I presented an uncomplicated, but highly reliable phone-controlled device — the 'Ring-A-Thing.' Connected to any phone line, the user would call, exactly allowing a pre-set number of rings to occur. Each was associated with turning an electrical device on or off, then immediately hanging up the phone. The controlled device would then be activated within a few seconds. The project presented here, is similar in purpose, but offers the ability to control multiple electrical devices through wireless means by pressing associated numbers via a Touch-Tone phone.

■The completed wireless control unit.

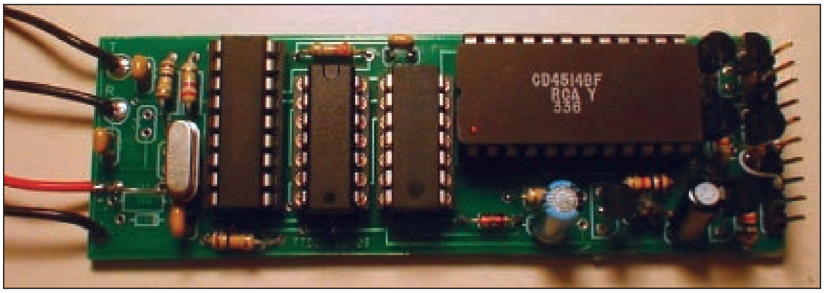
Since the advent of Touch-Tone [TT] phone service, there have been numerous 'control-related' projects designed using this medium, as well as other adaptations that we are well acquainted with — namely automatic phone answering systems with menus to navigate through or having personal data numerically entered, etc.

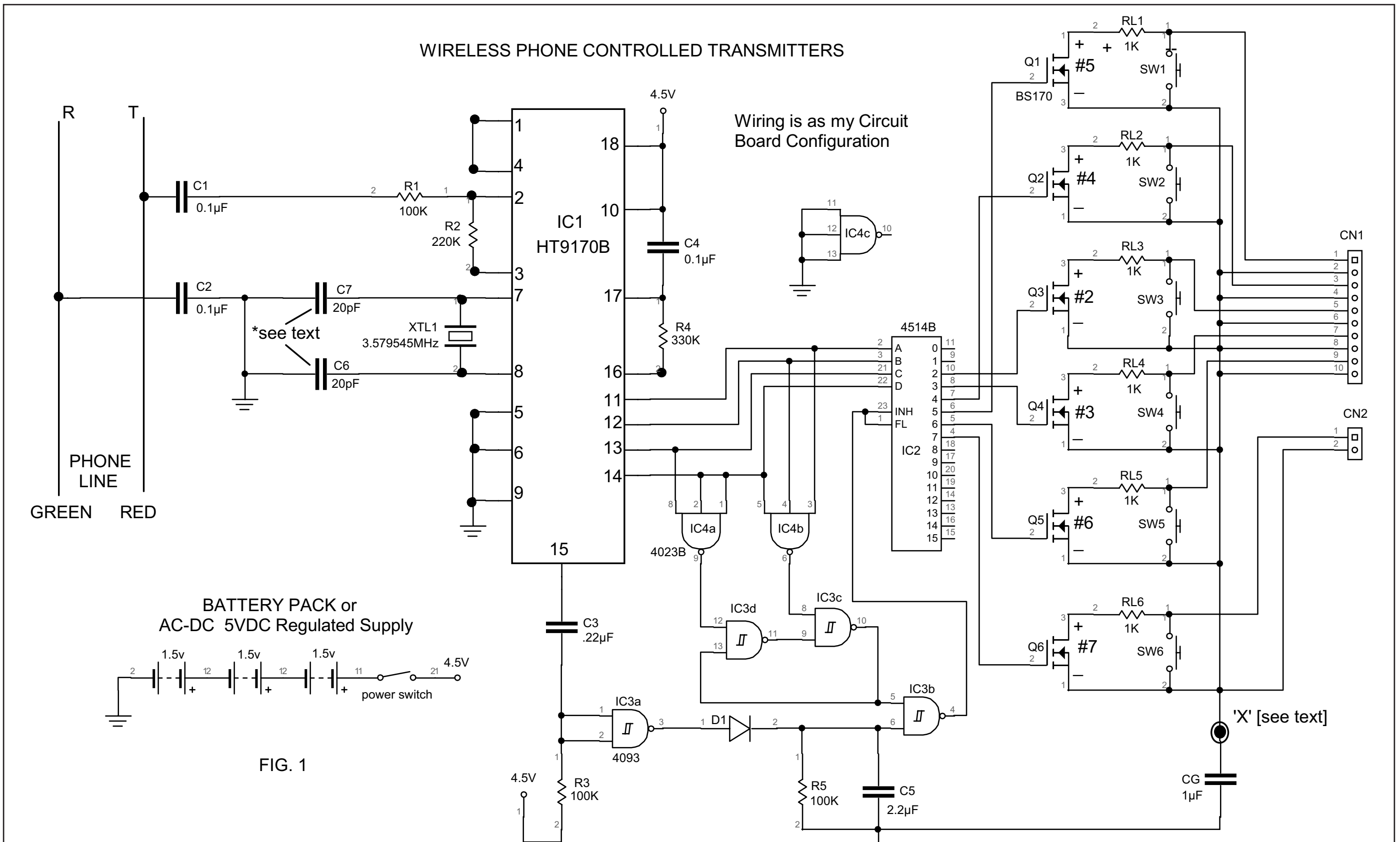
Using highly reliable TTs, I have designed several projects in the past to remotely and locally control everything from household lamps to high voltage equipment. With the use of a Touch-Tone generator IC, I assigned tones to devices that transmitted wind speeds and direction, wet conditions, water levels, broken fences, etc., over CB radio. The project in this article uses the phone line to activate wireless transmitters that control their associated receiver modules which are 120 VAC wall modules providing power to operate devices plugged into them (such as fans, radios, TVs, lamps, small motors, etc.). For devices with higher voltage and/or amp requirements, appropriate rated 120 VAC relays must be used.

Briefly, by definition, TT numbers are analog. For high reliability, each tone is comprised of two frequencies — one lower than the other as associated with their row and column format as it appears on a standard phone pushbutton keypad, from 0 to 9 with * and #. Some keypads required for special purposes allow for the A, B, C, and D inputs. This project requires only input numbers 2 to 7, as well as the * [#11] and # [#12] inputs. Although IC1 — the DTMF HT9107B TT decoder — will decode the A, B, C, and D inputs. (For Touch-Tone information, enter DTMF HT9170B into your Google search engine.)

A TT number is entered on a keypad, decoded to a binary-weighted number, and again to a single decimal equivalent number to activate a wireless transmitter [on and/or off] to control the electric device plugged into its associated AC wall module receiver. A transmitter switch is

■The controller circuit board.





■ SCHEMATIC 1. Project schematic.

electrically pressed by connecting the drain [+] and source [-] pins of each FET directly across the connections of each switch as indicated in **Schematic 1**. As each FET [Q1-Q6] is turned on by its respective IC2 output pin, it electrically closes the switch.

Although the transmitters indicated in this project have only one switch for on and off activation to their receivers, some models have wireless transmitters with both switches. If these latter-type transmitters are used, then wire each switch to a separate output FET for a total of three transmitters, i.e., Q1 on/Q2 off; Q3 on/Q4 off; Q5 on/Q6 off [refer to schematic].

Notice that IC1 is also capable of decoding all 16 TT tones [0-9; *, #, A, B, C, D] and output each as a 'binary-weighted' [8, 4, 2, 1] number from 0 to 15. Each of the four binary output pins are also wired directly into IC2, a 4514 [1-of-16] binary decoder which outputs the decimal number equivalent to the original TT input number. For example, a single TT input to IC1 for the number 9 would contain two frequencies: 852 Hz for low and 1,477 Hz for high. The two output binary bits [pin 11 and pin 14] are, respectively, 1 and 8 = 9, with both wired to the associated binary input pins of IC2 which momentarily causes ONLY the number 9 output [pin 17] to go high, (leaving all other output pins low).

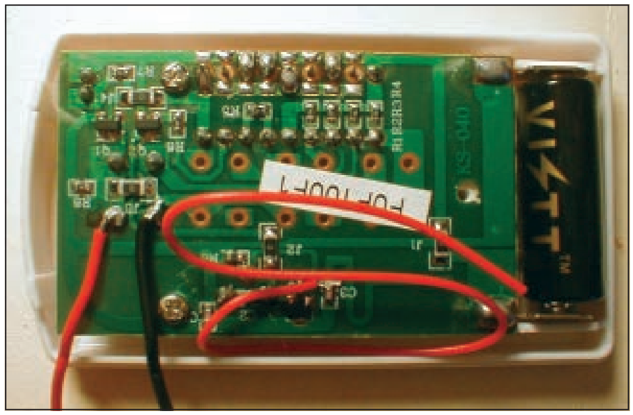
Referring to **Schematic 1** again, IC4 contains three triple-input NAND gates. Two gates are directly wired to both the IC1 output and IC2 input pins associated with the binary numbers of 11 and 12 which are, respectively, the * and # on the phone keypad. IC4b output pin 6 goes low when the * [11] key is pressed; IC4a output pin 9 goes low when the # [12] is pressed. IC3 contains four two-input NAND gates. IC3c and IC3d are wired in a 'flip-flop' configuration, being set or reset as per the low outputs of IC4a or IC4b.

When the * key is pressed [binary 11], IC4b output pin 6 goes low and sets IC3c output pin 10 to a high state. This enables the IC3b gate at input pin 5. To reset the flip-flop, the # key is pressed [binary 12], providing a low state to input pin 12 of IC3d. Output pin 15 of IC1 is normally at a low state. When any number is pressed on the keypad, pin 15 goes — and remains — high for as long

■ Super Switch.

as the key is pressed. In this condition, the R3/C3 one-shot timing circuit is not charged.

■ Wireless unit.



■ Transmitter switch wiring.

Gaining Control

To enable control of the wireless transmitters via IC2, the * [11] key must be pressed first and, when released, pin 15 immediately goes low again. This action charges the C3/R3 timing circuit, providing a low pulse [approximately .0147 s] to input pins 1-2 of IC3a. It's inverted to high again at output pin 3 and into the D1/R5/C5/IC3b timing network. It is then wired to input pin 6 which — when enabled by the flip-flop being set at input pin 5 — causes output pin 4 of IC3b to go into an enabling low state on IC2 at pins 1-23. This causes the respective output pin number to go momentarily high for the charge time set by R5xC5x.67, or approximately 150 ms. At this point, IC2 output pin 19 [#11] goes high but it is not connected to anything because the * input is only used to set the flip-flop. It also allows all following number inputs to momentarily turn FETs Q1 to Q6 on to activate their associated wireless transmitter switches (which will be electrically pressed in the same manner).

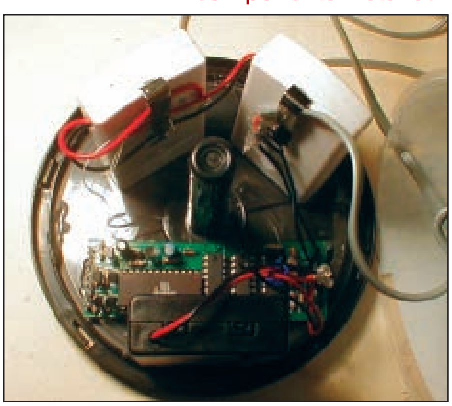
Resistors RL1-RL6 provide a load across the FETs and switches to prevent shorting. All source pins on the FETs are wired together to capacitor CG, providing a common ground connection between the wireless transmitters and circuit board power. Immediately following the IC3b timing period discharge, output pin 4 returns high again and all output pins to IC2 will go low. To disable any further inputs to IC2, the # key is pressed, resetting the flip-flop so that output pin 4 of IC3b remains high on IC2 input pins 1-23. This inhibits all further outputs in this condition.

That's about it. Using a circuit board is the ideal way to construct this project. Most wireless transmitter/receiver units of this sort provide an operating distance from 50 to 100 feet. As mentioned previously, there are various manufacturers, with transmitters having either one or two switches. The units used here are the "Super Switch 3," which can be purchased through Northern Tool Co. (**www.northerntool.com**; part #9719052). They offer a variety of frequencies for using multiple units, a single on/off switch action on both the transmitter and receiver wall module, and a 60 foot operating range.

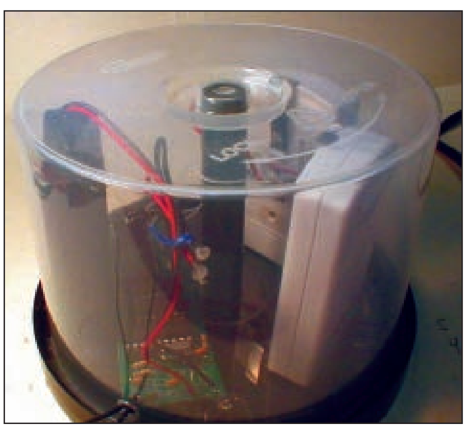
Putting It To Use

With this device attached to your phone line, you will have a handy controlling unit to operate multiple wireless transmitters from anywhere in the world. For example, if you are using a local cordless phone, simply press the talk button. When the dial tone is heard, press the * key to enable IC2, then press your transmitter activation number[s]. You will most likely hear an operator message but it won't interfere with your inputs. When finished, simply press the # key and hang up. For remote operation, you must have an answering machine connected to your phone line and then dial your phone number. As soon as the answering machine is done telling you to leave a message, press the * key, enter the transmitter number[s],

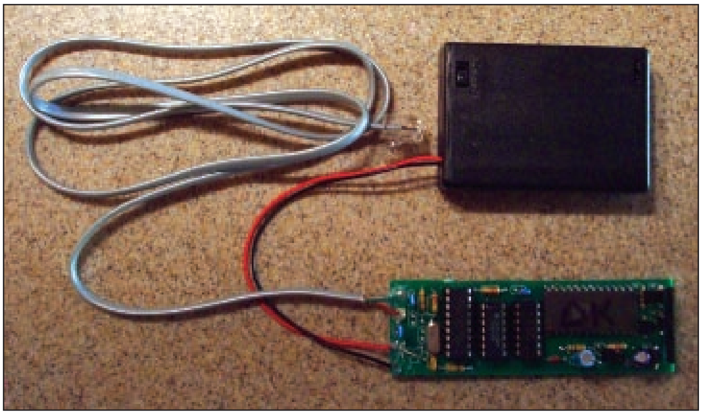
■ Lid from the container with the components installed.



■ Container with the clear lid on.



■ Lid on with decorative cover.



■ Available kit components.

then press the # key and hang up. (You will need to devise your own method of keeping track of the AC wall receiver on or off, positions. Whatever position you left it in after you pressed the # key will remain until it's changed by you.) Normal phone use is never disrupted and it will not change anything until the * key is pressed and the transmitters are activated again.

You can purchase an inexpensive cordless phone unit [Uniden brand works] and plug the controller directly into the base unit line input without being attached to the phone line. To do this, make sure that the RED phone line wire is connected to the T input [to capacitor C1] and DO NOT CONNECT the GREEN wire to anything. If it does not work, ONLY reverse the wires. Follow operating procedures outlined. If you are using it through the phone line, then connect both wires.

To be somewhat imaginative, I put two transmitters, the circuit board, and battery case into an empty CD container [see **photos**]. Using a glue gun, I dabbed glue all around the soldered wire connections and secured the other components on the clear plastic disk that's usually placed in with the CDs when you buy them. A hole was drilled in the CD lid for the phone line cord to pass through. I placed contact paper around the CD lid to be a little decorative.

■ Inside the enclosure.

Construction Hints

You can use the battery power supply configuration shown in **Schematic 1** or you can use an AC-DC regulated five-volt power supply. The battery holder mentioned in the **Parts List** has a switch built in and provides 4.5 volts using Duracell batteries. Mount the resistors and diode to the PCB first. Then solder in the four IC sockets [or ICs], followed by the capacitors and the crystal, not too close to the board.

(Note: IC1 manufacturer diagnostic specs have capacitors C6 and C7 indicated. Although I have provided for their installation on the circuit board, I have omitted them and the circuit works very well. Depending on your phone service, you may not need them installed and can eliminate them. Use them only if you are experiencing problems.)

Now, solder in the FETs, keeping the FET leads long and mounted high on the board. Lastly, connect the CN1 and CN2 headers, the phone line cord and the battery holder or power supply wires. Double-check the solder connections as you go along.

PARTS LIST

PART	SUPPLIER/PART#
IC1: HT9170B DTMF decoder	www.futurlec.com
IC2: 4514 4-bit 1-of-16 decoder	[Jameco #13522]
IC3: 4093 quad 2-input NAND gates	[Jameco #13400]
IC4: 4023 triple 3-input NAND gates	[Jameco #12845]
XTL: [3.579545]	[Jameco #14533]
IC 14-pin sockets (2)	[Jameco #526192]
IC 18-pin socket	[Jameco #683139]
IC 24-pin socket	[Jameco #683163]
R1, R3, R5: 100K	[Jameco #691340]
R2: 220K	[Jameco #691420]
R4: 330K	[Jameco #691657]
RL1-RL6: 1K FET load resistors	[Jameco #690865]
Q1-Q6: BS-170 FETs	[Jameco #256031]
C1, C2, C4: .1 µf	[Jameco #151116]
C3: .22 µf	[Jameco #33507]
C5: 2.2 µf	[Jameco #609561]
CG: 1 µf	[Jameco #330431]
*C6, C7: 20 pf [see text]	[Jameco #332322]
D1: 1N914	[Jameco #179207]
Power: 4.5V battery holder with switch	[Jameco #216144]
or five-volt AC-DC reg. supply	[Jameco #308700]
CN1: 10-pin male header	[Jameco #103393]
wire-wrap connections to transmitter switches;	
CN2: two-pin male header	[Jameco #879406]
wire-wrap connections to transmitter switches;	
RJ-11: Phone line/plug, 7 ft;	[Jameco #115588]
cut cord end; tin wires for circuit board connection	
Wireless Light Switches **www.northerntool.com**	[Part #9719052]

Misc: Wire-wrap wire and tool, solder equipment, wire, enclosure, project assembly material.

Circuit boards and kits are available through the *Nuts & Volts* website (**www.nutsvolts.com**).

Testing the Circuit Operation

If you're putting the project together without a circuit board, then all you will really need is a volt meter and a logic probe for the following test procedures. Determine where and how you want to mount the wireless transmitters, then remove the enclosures and the batteries. On the transmitter circuit board across the switch terminals, use your volt meter; one side will be ground [black wire] and the other will be positive [red wire].

As per **Schematic 1**, wire-wrap an appropriate length of wire onto the #1 CN1 header terminal and touch-solder the other end of the wire to the positive side of the transmitter switch. Prepare and wire-wrap another piece of wire of the same length from the #2 CN1 header terminal to the negative side of the transmitter switch. Double-check all wiring, replace the battery in the transmitter, and insert its associated receiver into an AC wall outlet, making sure it is set to off. Plug the phone line cord into a spare phone jack and then provide power to the controller circuit board. With a desk phone earpiece off hook or cordless handset talk button pressed, you should hear a dial tone. Press the * key and then the number 5. You may hear an operator message or other signal at this point, but it will not affect further testing.

If all the wiring is correct, then the transmitter LED momentarily flashes and the receiver should turn on. If nothing works, then check the soldering connections, and inputs and outputs on IC1 with a logic probe. If all is well, press the number 5 again; the receiver module should turn off. If so, press the # key to lock out the control function.

Now press the number 5 again and the receiver module should remain off. Press the * key, followed by the number 5, and the receiver should turn on. Press the number 5 once more to turn the receiver off, followed by pressing the # key to disable the remote control function. Test the other five outputs from IC2 to their respective transmitters.

Notice that all of the FET source pins to the transmitter switches are common to the positive side of capacitor CG [at X]. Depending on your assembly, you can wire each of the transmitter switch grounds together and have only one wire connected back to the circuit board at the X point.

Last Call

I tested this project by having someone call me with their cell phone from Florida and follow all the procedures mentioned. It worked without any complications, just as it did with on-premise and local phone calls. I also used it without the base unit of my cordless phone plugged into the phone line and I had no problems whatsoever. **NV**

Experiments with Alternative Energy

Part 7 - Experimenting with the Whirlybird Wind Turbine

By John Gavlik, WA6ZOK

Last time, I showed you how to build the Whirlybird™ three-phase AC wind turbine along with some background information on wind turbine technology. This time, I'll show you some of the interesting experiments that can be done with this device. Here are the topics I'll address:

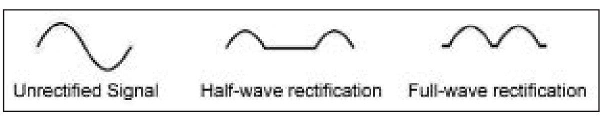
- Three-Phase Rectification
- Understanding Three-Phase Voltage Output
- Smoothing The Rectified DC Output
- Wind Speed, RPM, and Power Output
- The Effects of Magnetic Coupling
- Star versus Delta Windings
- Hands Off Braking

Most of these topics are really the titles of the experiments that I've developed around the Whirlybird wind turbine, and I'll be referring to them in some detail in this article. So, if you've already built your Whirlybird wind turbine and want to know more about how it works, this article is for you. If you are just thinking about building it, this may convince you to do so. Either way, you'll learn a good deal more about wind turbine technology. Let's get started.

Three-Phase Rectification

The kind of three-phase rectification that's done with the Whirlybird wind turbine is full-wave rectification as opposed to half-wave rectification (**Figure 1**). Remember, a diode conducts current in only one direction (from positive to negative) and blocks current flowing the other way. Half-wave rectification requires only one diode while full-wave rectification requires at least four. The major difference between a single phase and three phases is that we have three coil groups as compared to one. So, the question is "how does one effectively expand the four-diode, full-wave rectifier circuit to accommodate two more coils?" The simple answer is "add two more diodes." The question then becomes "how does that work?" Well, let's start with how a full-wave rectifier works and move up to a three-phase rectifier.

Figure 1. Half-Wave and Full-Wave Rectification.



In a typical single-phase full-wave rectifier (**Figure 2**), two of the four diodes conduct current during the positive portion of the AC sine wave (D1 and D4) while the other two

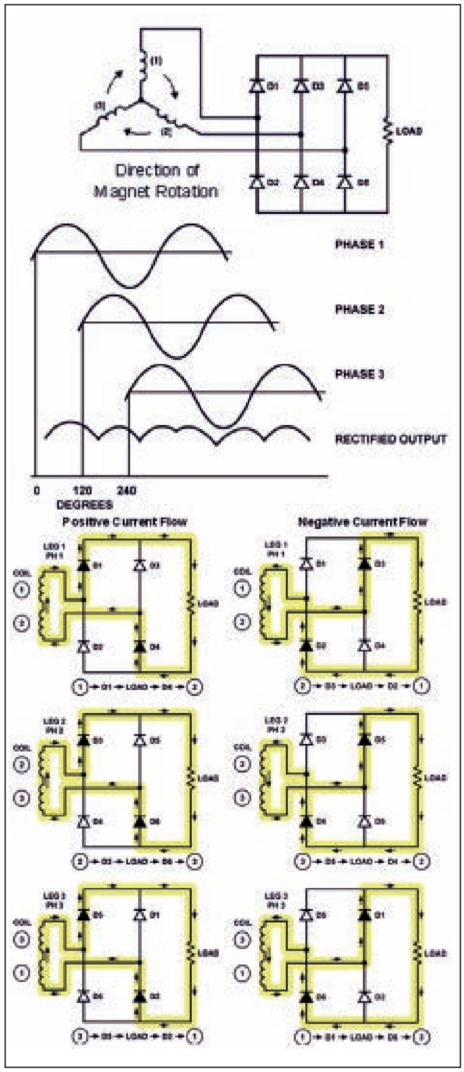


Figure 2. Three-Phase Full-Wave Rectifier Current Flow.

conduct current during the negative portion (D2 and D3). The result is that the negative portion of the sine wave is "flipped" to positive which provides double the power producing capability as compared with a half-wave rectifier. The diodes themselves require a minimum activation voltage or "drop" of typically 0.6 volts before current actually can begin to flow through them. This is called the forward-biased voltage (**Figure 3**) which accounts for the "flat," no voltage portion of the rectified voltage traces.

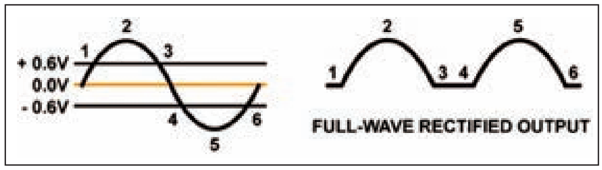
To extend this full-wave rectified scheme to three-phase, let's first consider the three coils that make up the three-phase circuit. For simplicity, I'll use the Star configuration although the Delta configuration works exactly the same way. **Figure 2** shows the three coils grouped as three "legs" with two coils per leg, so mentally treat each leg as a voltage source that generates a sine wave; only each sine wave is displaced [in time] from the others based on the 120 angular degree coil separation.

To make things simpler, I've made the leg numbers correspond to the phase numbers 1, 2, and 3. By studying the phase relationships for the three legs, I hope you can see how the technique works for all three legs with just two extra diodes. If you don't "get it" at first, look again; it's important in regards to what you will learn about next.

Understanding Three-Phase Voltage Outputs

This is where I'll get into interfacing the Parallax BS2 or

Figure 3. Effect of Diode Voltage Drop in Full-Wave Rectified Voltage.



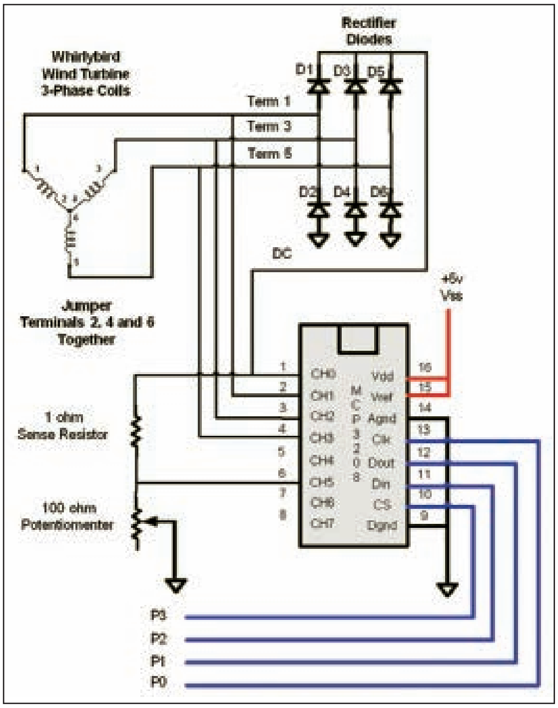
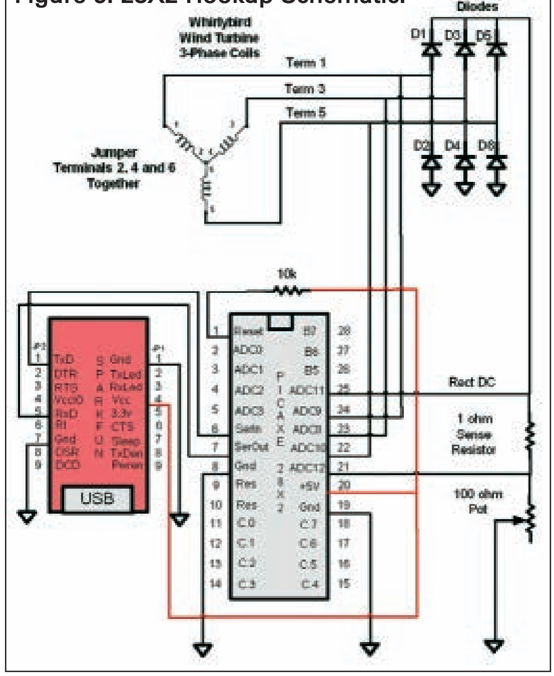


Figure 4. BS2 Hookup Schematic.

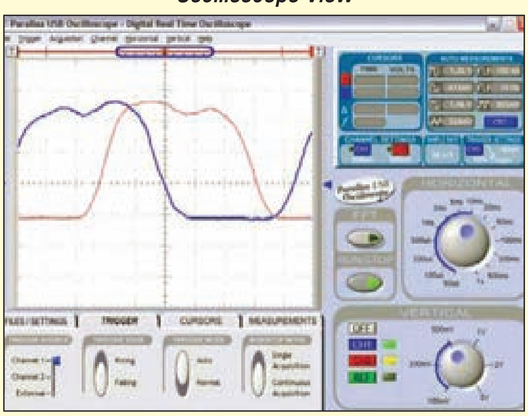
Figure 5. 28X2 Hookup Schematic.



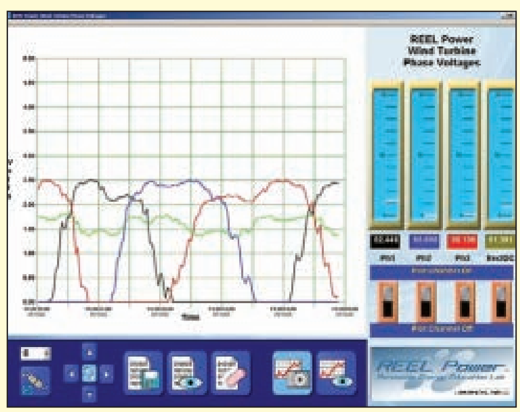
Why are the Phase Waveforms Jagged?

The "clean" rectified sine waves depicted on the left are like those that are actually produced by the wind turbine's full-wave rectifier circuit. The jagged nature of the waves in the plots on the right is due to the rather coarse sampling of the fundamentally smooth waves — about one millisecond per sample. The micro captures a series of samples of different waves to construct the plotted image which accounts for the variance in the voltage plots, since each wave is not exactly the same as the one before it or after it in terms of voltage level. If the microcontroller code were as fast as an oscilloscope, the plots would look like those on the left. However, since we're only using Basic interpreters as opposed to compiled code in both the BS2 and chips, this is not possible.

Oscilloscope View



Software Plot View



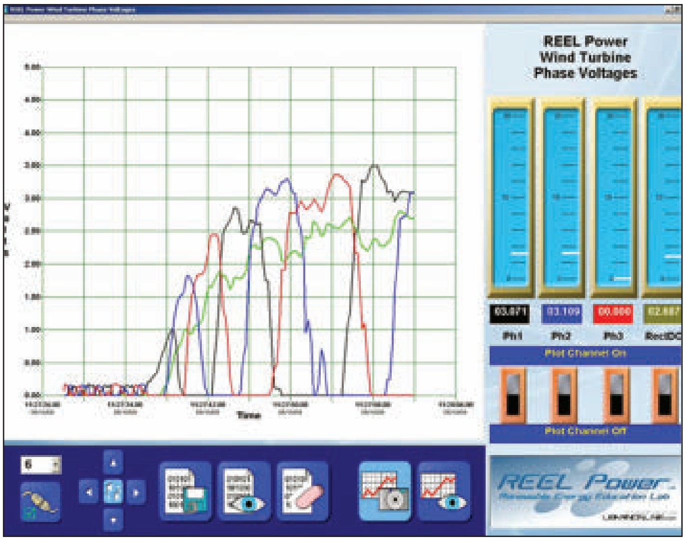


Figure 6. Whirlybird Spinning Up to Speed.

PICAXE 28X2 microcontrollers to the wind turbine. **Figures 4** and **5** show the schematic diagrams for attaching the Whirlybird wind turbine to the BS2 and 28X2, respectively. The A/D converter inputs pick up the three independent voltage phases after they are rectified along with the rectified DC voltage. The rectified DC voltage is routed through the one ohm current sense resistor into a 100 ohm potentiometer load. This way, we can measure the effects of variable loading on the turbine, as well as learn about the three-phase voltages. For the most part, these two schematics will suffice for the rest of the discussion in this article.

In **Figure 6**, you can see the computer plot of the three-phase voltages (black, blue, and red) along with the rectified DC voltage (green) as the turbine spins up to speed. At this point, the display will look confusing and cluttered; clicking the graphic icon switches under each of the phase meters turns off the phase voltage plots to display just the rectified DC by itself (**Figure 7**). As you can see, this is not a smooth DC signal but one that follows the voltage peaks of the three phases. Nevertheless, it looks far less confusing without the clutter of the other three-phase plots.

To get a better idea of how the rectified DC signal is created, **Figure 8** shows only a rectified Phase 1 (black line) along with the DC signal (green line). Here you can witness how the DC signal follows the peak of this phase voltage, riding just below it due to the 0.6 volt rectifier diode drops just discussed. **Figure 9** shows Phase 2 (blue line) added to the plot where it begins to become even clearer as to how the rectified DC signal follows the phase voltage peaks. Finally, **Figure 10** shows all three phases along with the rectified DC voltage. Seeing it one phase plot at a time generally makes more sense than seeing all the signals plotted at once the first time.

It should also be noted that while the plots of the phase voltages appear jagged, they are actually smooth (see **sidebar**). The jaggedness is due to the firmware sampling technique (**Figure 11**) that takes individual voltage samples of different parts of the phase waveforms over a period of several milliseconds. Unlike an oscilloscope that captures the waveforms all at once, our slower BS2 and 28X2 micros must take discrete samples of each of the phases at various times; mainly because they are both programmed in the interpretative Basic language and not the faster C or assembly languages. Each discrete sample consisting of all three phases and the rectified DC voltage is then transmitted to the computer to create the waveforms you see plotted.

Smoothing the Rectified DC Output

As **Figure 12** illustrates, by adding a 330 µf capacitor across the 100 ohm load the rectified waveform smoothes out considerably — not exactly pure DC but getting there. When the input voltage is higher than the capacitor's voltage (at the moment), the rectified DC charges the capacitor to the peak DC level. When the input voltage falls below the capacitor's voltage, the capacitor discharges its stored energy into the load. Thus, the capacitor helps to stabilize the rectified voltage; so adding more capacitance is a way to make the rectified DC smoother.

Figure 7. Rectified DC Wave Only.

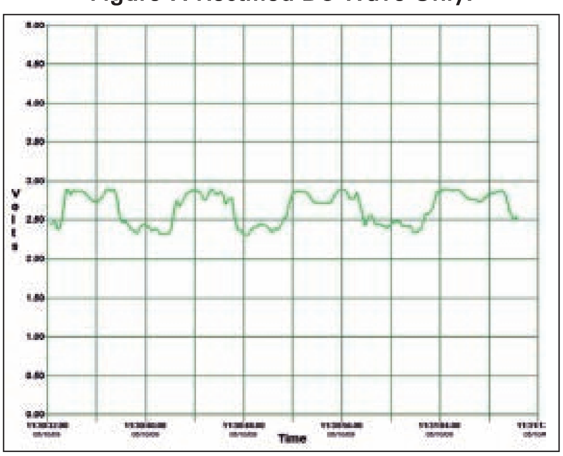
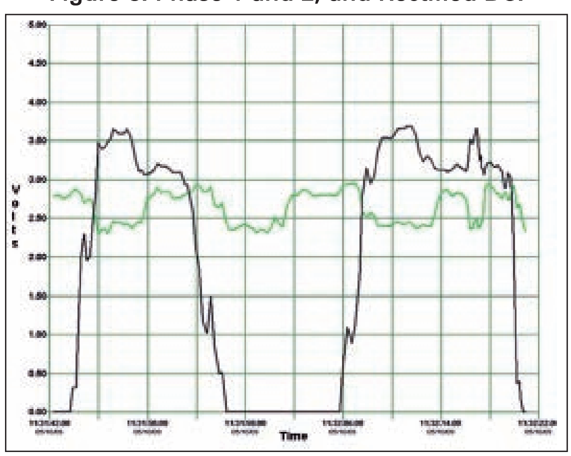


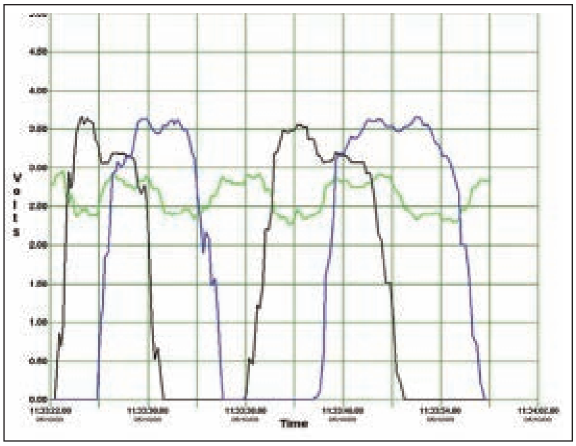
Figure 8. Phase 1 and 2, and Rectified DC.



Wind Speed, RPM, and Power Output

As one would expect, wind speed affects the RPM and DC voltage levels; the greater the wind speed, the greater the RPM and DC

Figure 9. Phase 1 and 2, and Rectified DC.



voltage, and vice versa. What is most interesting, however, is the effect of the wind force itself. In general, if the wind speed is doubled the output power increases by eight times — the cube of the wind speed. The other main variable is the blade area or — in our case — the area of the turbine fan that captures the wind. These and other variables are expressed by the following equation for wind turbine output power:

$$P = 0.5 * ? * A * V^3 * E$$

Where:
P = Power in Watts
ρ = Air Density in Kg/m^3 (about 1.225 Kg/m^3 at sea level, less higher up)
A = Rotor Swept Area in $m^2 = \pi r^2$ (r = radius of the rotor or fan)
V = Wind Speed in m/s (cubed)
E = Efficiency in percent

If the radius of the rotor blades is doubled, the swept area is quadrupled. If the wind speed is reduced by half, the power is reduced to 1/8 of the original power. Efficiency is the wild card in the equation where many other factors go into the unique design of each wind turbine.

This equation was derived not too long ago by Albert Betz (1885–1968) who was a German physicist and a pioneer of wind turbine technology. Betz found out that we can only harvest [at a maximum] 16/27 or 0.593 of the power from the wind. This number is called the Betz coefficient and is the theoretical maximum efficiency that a turbine can harvest from the wind.

You can validate this equation by using an inexpensive hand-held anemometer to measure the turbines wind speed. The one I've chosen can measure wind speed in meters per second as called for by the above equation (**Figure 13**). With it, you can set up the turbine in front of a large table fan and measure the voltage, current, and power at different fan speed settings and distances from the turbine (**Figure 14**). Replace the wind turbine with the anemometer and measure the average wind speed at each electric fan speed setting by adjusting the anemometer position at left, center, and right. With both the wind speed and the measured power, you can plug in the air density and rotor swept area numbers and compute the efficiency, E, at different wind velocities. You'll see the efficiency is quite low, mainly because the Whirlybird is primarily a "drag" type as opposed to a "lift" type (typically seen in models with curved blades that act like an airplane wing). We'll cover more about this type of wind turbine in the next article. Regardless, this is a great way to understand the relationship between wind speed, RPM, and output power.

The Effects of Magnetic Coupling

Probably as significant as wind speed — or maybe more so — magnetic coupling is an important consideration in every wind turbine's design. In effect, the closer the rotating magnets are to the coils, the more power that will be generated. The Whirlybird design is ideal for adjusting and measuring this effect; here's what you can do.

First, set the wind turbine in front of a large table fan and let it ramp up to speed. Adjust the potentiometer load to any desired resistance and measure the power. Then, stop the fan and disassemble the turbine into the rotor and stator halves (refer to Part 6 if this is unclear). Next, add a single flat washer as an additional spacer between the rotor and stator

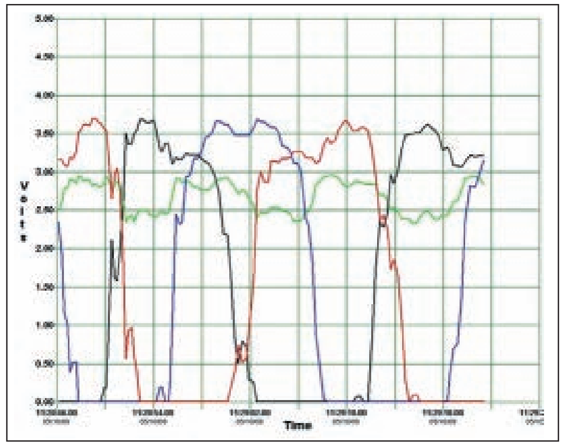


Figure 10. Phase 1, 2, and 3, and Rectified DC.

Figure 11. Phase Sampling Technique.

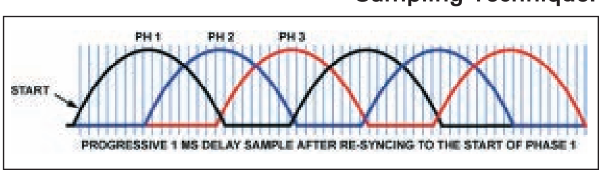
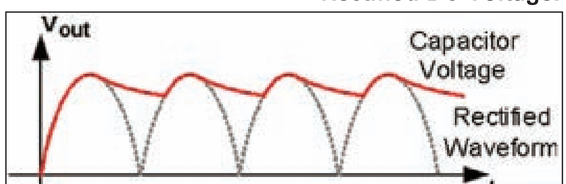


Figure 12. Capacitor Filtered Rectified DC Voltage.



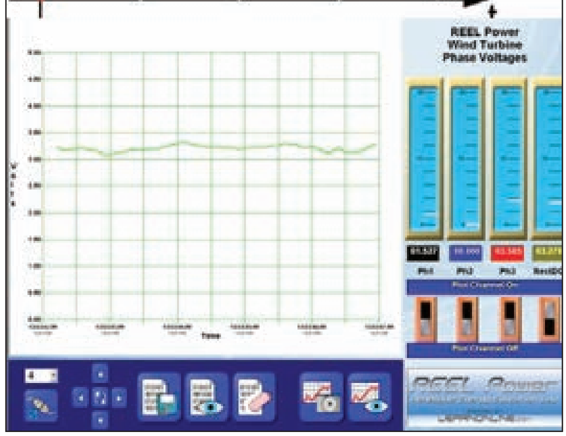


Figure 13. Hand-held Anemometer.

Coils	Volts	Amps	Watts	RPM
Star	2.671	0.027	0.072	119
Delta	1.730	0.051	0.088	119
Star	1.834	0.017	0.031	92
Delta	1.187	0.026	0.031	92

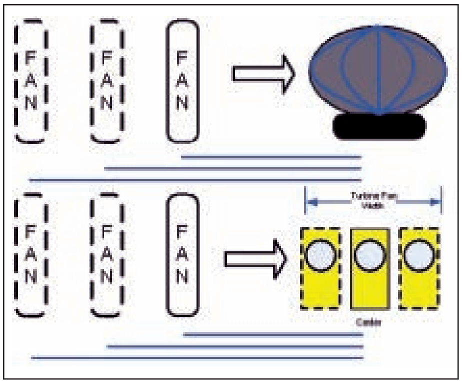


Figure 14. Fan and Anemometer Positions for Wind Experiment.

assembly (**Figure 15**). Repeat the same measurement with the same load resistance, wind speed, and distance between the table fan and wind turbine.

I'm sure you'll be amazed at how much power is lost just by adding the amount of space from a single flat washer between the coils and rotating magnets. If you've built the wind turbine and haven't yet optimized it by finding the minimum number of flat washer spacers necessary for smooth running, do so now; it will improve its performance greatly.

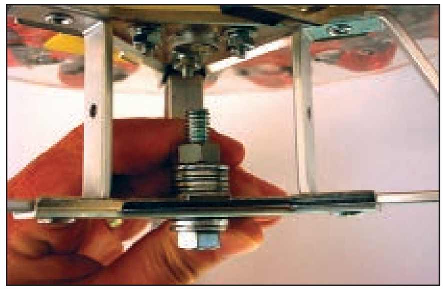


Figure 15. Adding Washer Spacers.

Star versus Delta Windings

You learned in Part 6 that the three-phase coils can be wired in two distinct arrangements: Star and Delta (**Figures 16** and **17**). While both coil arrangements produce the same power, the Star generates more voltage than current; the Delta configuration generates more current than voltage for any given RPM setting. From the experiment of the same name, here are the data I took with the wind turbine at two RPM settings. Volts, amps, watts, and RPM are all displayed on the REEL Power software (which is where I got it).

While the power values for the faster speeds are not exactly the same, they are close enough to illustrate the point. So, what is it about the Star and Delta windings that makes this difference in the voltage and current production? Leg resistance is the key!

The basic reason all of this is true is because the total coil resistance for each leg is different when wired in either a Star or Delta configuration. Assuming that each coil is wound with the same wire thickness and number of turns, the coil resistance for each of the three coils will be the same. However, the total resistance per leg is much different when wired in either a Star or Delta arrangement.

In the Whirlybird wind turbine, each "coil" measures about 26 ohms which is the sum of the four individual coils that are wired together in series as shown in **Figure 18**. Therefore, when we speak about coils here, assume that we mean the "coil group" as a whole. For the following discussion, refer to the Star

(**Figure 16**) and Delta (**Figure 17**) diagrams. In the Star arrangement, the resistance of each leg is the sum of two coils or about 52 ohms.

```
Leg 1 = coil 1 + coil 2 = 26 +
26 = 52 ohms
Leg 2 = coil 2 + coil 3 = 26 +
26 = 52 ohms
Leg 3 = coil 3 + coil 1 = 26 +
26 = 52 ohms
```

In the Delta arrangement, the resistance of each leg is the parallel equivalent of one coil in parallel with the sum of the other two coils. The leg resistance is about 17 ohms by the following:

```
Leg 1 = coil1 || (coil2 +
coil3) = 26 || 52 = 17.3 ohms
Leg 2 = coil2 || (coil1 +
coil3) = 26 || 52 = 17.3 ohms
Leg 3 = coil3 || (coil1 +
coil2) = 26 || 52 = 17.3 ohms
```

where || indicates "in parallel with."

Using the formula for two resistors in parallel, we have:

```
R = (R1 * R2) / (R1 + R2)
R = (26 * 52) / (26 + 52)

R = 1352 / 78

R = 17.3 ohms
```

Therefore, with a smaller resistance more current can flow in the Delta arrangement.

Since Star and Delta wirings both deliver the same power, does it matter which is used? It all depends on the application. On a commercial wind turbine, an alternator starts off wired in Star and then at a certain RPM switches to Delta. Most large commercial wind turbine generators do exactly this — they switch from Star to Delta at the right time using sophisticated electronics and software to sense when to perform the switch. High winds are another reason to switch to Delta, since doing so presents a heavier load (lower resistance) which slows the blade rotations.

Hands-Off Braking

I just said that switching to Delta slows down the rotation. So, how does this happen?

With a lower coil resistance coupled with (we can assume) a fixed load resistance, the overall power requirement now goes up for a given wind speed. With a constant wind speed, this means that the wind turbine cannot extract any more wind power to supply the heavier load, so something has to give. What gives is the rotational speed of the blades or the fan (in our case). It's just like riding a bicycle on level ground then going up hill. This is slowing your peddling rate because you — the power source — only have so much power to supply.

You can see this happen when you vary the potentiometer from its highest resistance to zero. The resistor will act like a brake for the rotating fan by slowing it as the resistance decreases. If the wind speed is light enough, it will even stop the turbine from spinning completely. If you haven't already tried this, do so and watch the RPM levels change with the resistance change. Keep in mind the power is going into heating the resistor load. However, with our small wind powerhouse it won't create any real hazard. A commercial wind turbine can destroy itself if this same condition is allowed to happen.

Summary

This time, you've learned more of the details surrounding the theory and operation of the Whirlybird wind turbine and why three-phase power is the logical choice for commercial power generation. The Whirlybird is a vertical axis type turbine (VAWT) and its fan is really a "drag" mechanism for capturing the wind. In other words, the fan itself does nothing to amplify the wind's power; it just allows itself to be pulled around by the blowing wind.

Next time, I'll continue our wind series and introduce you to a model sized, three-phase Horizontal Axis Wind Turbine (HAWT) that uses propeller blades that provide lift like an airplane wing.

As a matter of fact, the blades of this three-phase turbine are designed to aircraft blade standards. And like a modern aircraft, you can change the pitch of the blades. I'll do even more experiments with this unique HAWT. In the meantime, you can view all the experiments for the Whirlybird wind turbine at **www.learnonline.com** → Experimenter Kits.

Until next time, conserve energy and "stay green." **NV**

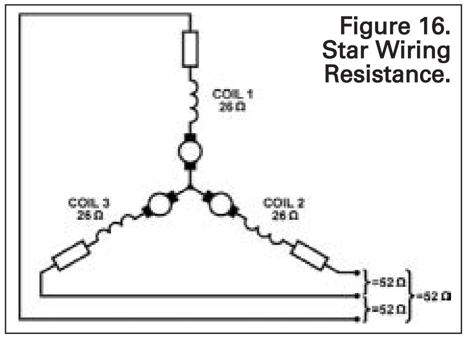


Figure 16. Star Wiring Resistance.

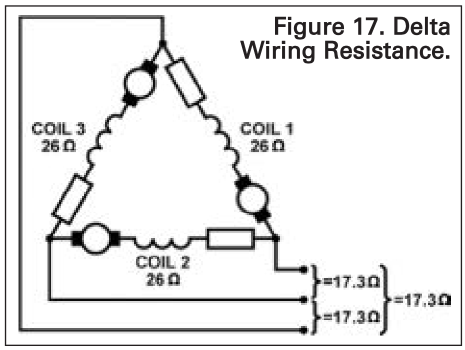
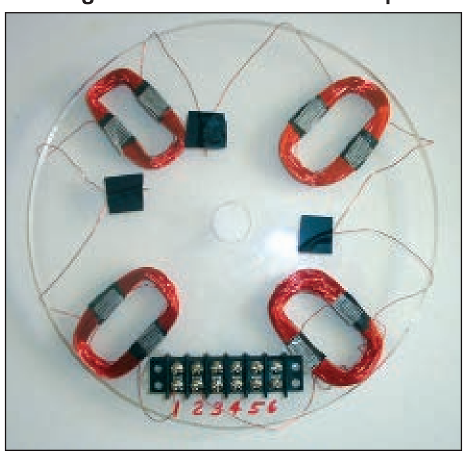


Figure 17. Delta Wiring Resistance.

Figure 18. Phase 1 Coil Group.



Serial Communications Part 2 – A Simple Terminal

Follow along with this series! Joe's book & kits are available in our webstore at **www.nutsvolts.com**

by Joe Pardue

Recap

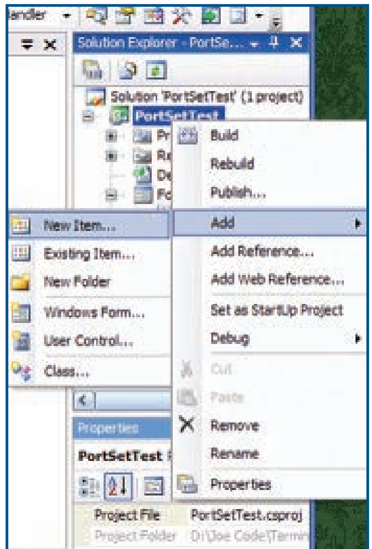
Last month, we introduced you to serial communications between a PC and an Arduino using C# or Visual Basic Express .NET (both free from **www.microsoft.com/express**/. (The article shows the C# examples and the Visual Basic examples are in the Workshop19.zip). This month, we will look at selecting a serial port, getting user input, and then we will expand on all this to build a simple terminal.

Getting User Input

In this section, we will create a dialog form to allow the user to select an available serial port and set the UART parameters needed for the communication link (Baudrate, Data Bits, Parity, Stop Bits, and Handshaking). To create and test the Settings dialog, we will use a test form — PortSetTest — to call it. Later, we will access this form by clicking the settings menu item from the Simple Terminal (next month, we will use it for a Arduino voltmeter program).

- Open either C# Express and create a new project named PortSetTest.
- Create a form to look like **Figure 1**.
- The text list is made from six labels using the text shown.
- In the Solutions Explorer, right-click 'PortSetTest' and from the drop-down menus, select Add/New Item as shown in **Figure 2**.
- In the Add New Item form shown in **Figure 3**, highlight 'Windows Form,' change the name to 'PortSettings.cs' if C#, and click Add.
- The blank form will open.
- In the Properties Window, select 'FormBorderStyle' and select 'FixedDialog.' This type of form can post a DialogResult message that the calling form — in this case, the Simple Terminal form — can process (don't panic, this is all done for you and is simpler than it sounds). We will use the DialogResult to learn if the user clicked the Settings form Okay or Cancel button.
- Now you should be able to do some stuff without a lot of hand-holding:
  - — Change the form name to 'Setting.'
  - — Add the Smiley.ico icon.
  - — Change the form color to Bisque.
- Add the controls shown below:
  - — The upper white box with listBoxPorts is a listBox which you will name listBoxPorts.
  - — The control with the label 'Baudrate' is a ComboBox; change its name to comboBoxBaud.
- Your Settings form should look like **Figure 4**.
- Add the DevInfo file to the project.
- Copy the DevInfo.cs (if C#) to the directory containing the PortSettings project files. You will find the DevInfo file in the Workshop19.zip.
- Refer to the Add/New Item image as shown in **Figure 2**.
- Open the same menus in the Solution Explorer, but instead of selecting Add/New Item, select Add/Existing Item and select DevInfo.cs.
- Note that we are intentionally not using the SerialPort GetPortNames function since it doesn't work correctly. (Microsoft released a buggy function — hard to believe, isn't it?) We will not look at this file in detail since it

■ FIGURE 1. PortSet Test.

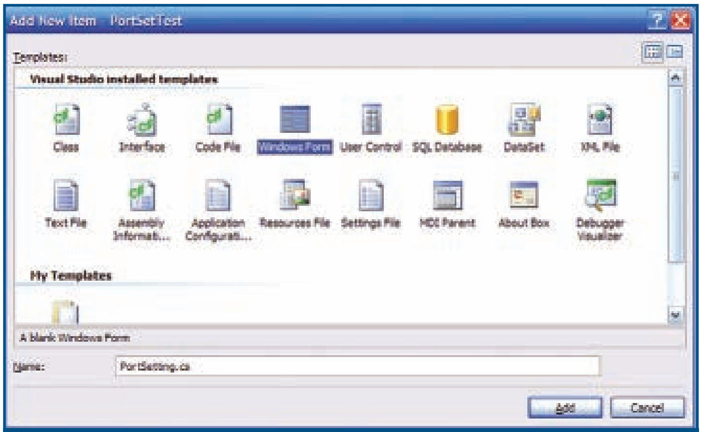


■ FIGURE 2. Add New Item.

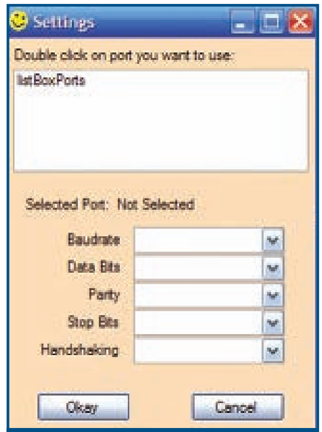
contains some advanced concepts, but we will use it to get the names of the available ports. The source code is in Workshop19.zip. If you want to learn the gory details, they are contained in *The Virtual Serial Port Cookbook* available from *Nuts & Volts* and Smiley Micros.

- In C#, add to the 'using' list at the top of the program:

```
using DevInfo;
```



■ FIGURE 3. Add New Item Form.



■ FIGURE 4. Settings Form.

- In C#, add to the constructor:

```
public PortSettings()
{
     InitializeComponent();

     // Get a list of the Serial port names
     string[] ports = GetPorts();

     int i = 0;
     foreach (string s in ports)
     {
          if (s != "")
          {
               listBoxPorts.Items.Insert(i++, s);
          }
     }

     // Set first Serial port as default
     GetCOM(0);

     // Initialize baudrates in combobox;
     comboBoxBaud.Items.AddRange(new object[]wrap
     {"75","110","134","150","300","600",wrap
      "1200","1800","2400","4800",
      "7200","9600",wrap
      "14400","19200","38400",
      "57600","115200","128000"});

     // Set Handshaking selection
     // We will only use these handshaking types
     comboBoxHandshaking.Items.Add("None");
     comboBoxHandshaking.Items.Add("RTS/CTS");
     comboBoxHandshaking.Items.Add("Xon/Xoff");

     // Set Parity types
     foreach (string s in
     Enum.GetNames(typeof(Parity)))
     {
          comboBoxParity.Items.Add(s);
     }

     // Set Databits
     // FT232R UART interface supports only 7
     // or 8 data bits
     // comboBoxDataBits.Items.Add("5");
     // not supported
     // comboBoxDataBits.Items.Add("6");
     // not supported
     comboBoxDataBits.Items.Add("7");
     comboBoxDataBits.Items.Add("8");

     // Set Stopbits
     // FT232R UART interface supports only 1
     // or 2 stop bits
     // comboBoxStopBits.Items.Add("None");
     // not supported
     comboBoxStopBits.Items.Add("1");
//comboBoxStopBits.Items.Add("1.5");
     // not supported
     comboBoxStopBits.Items.Add("2");

     comboBoxBaud.Text = "19200";
     comboBoxParity.Text = "None";
     comboBoxDataBits.Text = "8";
     comboBoxStopBits.Text = "1";
     comboBoxHandshaking.Text = "None";
}
```

- Note the word '***wrap***.' This means that the prior line of code was too long to display so it was wrapped to the next line for the available space and not wrapped for the actual source code. We will use this as a convention throughout our tutorial. If you enter '***wrap***' in your code, you'll get an error. (If you think I'm over-explaining this, you haven't seen my email inbox.)
- From the Settings Designer form, click the Okay button and the Cancel buttons, and then in the Settings text editor window add the following DialogResults to the functions (in C#,):

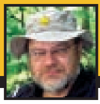
```
private void buttonOkay_Click(object
sender, EventArgs e)
{
     DialogResult = DialogResult.OK;
}

private void buttonCancel_Click(object
sender, EventArgs e)
{
     DialogResult = DialogResult.Cancel;
}
```

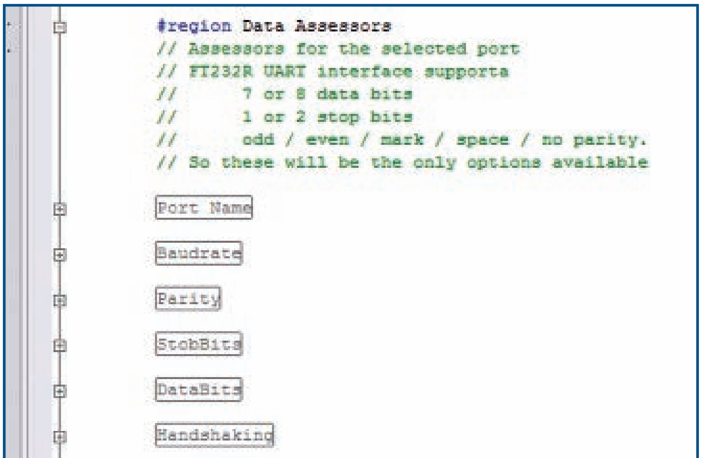
These functions close the form and post the DialogResult message for the PortSettingsTest form which called the Settings form.

In the source code in Workshop19.zip, you will see #region and #endregion. The IDE uses #region and #endregion to show or hide code sections. When you click on the + or - to the left #region, the code section collapses or expands. We use these to simplify the code visually, making it easier to follow the overall code logic. Collapsing the regions looks like **Figure 5**.

Create data assessors and indexChanged functions by clicking the 'SelectedIndexChanged' event in the events section (lightning bolt) of the Properties window.

In C#, add:





■ FIGURE 5. Regions.

```
#region Data Assessors
// Data assessors and index changed functions
// FT232R UART interface supporta
//      7 or 8 data bits
//      1 or 2 stop bits
//      odd / even / mark / space / no parity.
// So these will be the only options available

#region Port Name
// Assessor for the selected port name
private string SelectedPort = "";
public string selectedPort
{
    get
    {
        return SelectedPort;
    }
    set
    {
        SelectedPort = value;
        labelPort.Text = "Selected Port =
        " + SelectedPort;
    }
}
#endregion

#region Baudrate
private int SelectedBaudrate;
public int selectedBaudrate
{
    get
    {
        return SelectedBaudrate;
    }
    set
    {
        SelectedBaudrate = value;
        comboBoxBaud.Text = value.ToString();
    }
}

private void
comboBoxBaud_SelectedIndexChanged(object sender,
EventArgs e)
{
    selectedBaudrate = wrap
Convert.ToInt32(comboBoxBaud.Items[comboBoxBaud.
SelectedIndex]);
}
#endregion

#region Parity
private Parity SelectedParity;// = Parity.None;
public Parity selectedParity
{
    get
    {
        return SelectedParity;
    }
    set
    {
        SelectedParity = value;
        comboBoxParity.Text = value.ToString();
    }
}

private void
comboBoxParity_SelectedIndexChanged(object
sender, EventArgs e)
{
    string temp =
comboBoxParity.Items[comboBoxParity.SelectedIndex
].ToString();

    switch (temp)
    {
        case "Even":
            selectedParity = Parity.Even;
            break;
        case "Mark":
            selectedParity = Parity.Mark;
            break;
        case "None":
            selectedParity = Parity.None;
            break;
        case "Odd":
            selectedParity = Parity.Odd;
            break;
        case "Space":
            selectedParity = Parity.Space;
            break;
        default:
            selectedParity = Parity.None;
            break;
    }
}
#endregion

#region StopBits
private StopBits SelectedStopBits =
StopBits.One;
public StopBits selectedStopBits
{
    get
    {
        return SelectedStopBits;
    }
    set
    {
        SelectedStopBits = value;
        comboBoxStopBits.Text =
        value.ToString();
    }
}

private void
comboBoxStopBits_SelectedIndexChanged(object
sender, EventArgs e)
{
    string temp = wrap
comboBoxStopBits.Items[comboBoxStopBits.Selected
Index].ToString();

    switch (temp)
    {
        case "None":
            selectedStopBits = StopBits.None;
            break;
        case "1":
            selectedStopBits = StopBits.One;
            break;
        // case "1.5": // not supported by
                       // FT232R
            // SelectedStopBits =
            // StopBits.OnePointFive;
            //break;
        case "2":
            selectedStopBits = StopBits.Two;
            break;
        default:
            selectedStopBits = StopBits.One;
            break;
    }
}
#endregion
```

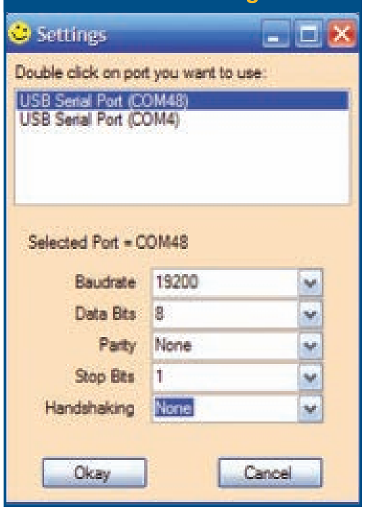
```
#region DataBits
private int SelectedDataBits = 8;
public int selectedDataBits
{
    get
    {
        return SelectedDataBits;
    }
    set
    {
        SelectedDataBits = value;
        comboBoxDataBits.Text =
        value.ToString();
    }
}

private void
comboBoxDataBits_SelectedIndexChanged(object
sender, EventArgs e)
{
    if (comboBoxDataBits.SelectedIndex == 0)
    selectedDataBits = 7;
    else selectedDataBits = 8;
}
#endregion

#region Handshaking
// We will only use None, Xon/Xoff, or Hardware
(which is RTS/CTS)
private Handshake SelectedHandshaking =
Handshake.None;
public Handshake selectedHandshaking
{
    get
    {
        return SelectedHandshaking;
    }
    set
    {
        SelectedHandshaking = value;
        comboBoxHandshaking.Text =
        value.ToString();
    }
}

private void
comboBoxHandshaking_SelectedIndexChanged(object
sender, EventArgs e)
{
    if (comboBoxHandshaking.SelectedIndex == 0)
    selectedHandshaking = wrap
Handshake.None;
    else if (comboBoxHandshaking.SelectedIndex
    == 1) selectedHandshaking = wrap
Handshake.RequestToSend;
    else if (comboBoxHandshaking.SelectedIndex
    == 2) wrap
selectedHandshaking = Handshake.XOnXOff;
    else selectedHandshaking = Handshake.None;
}
#endregion
```

- We will test the Settings dialog by displaying the data in the PortSetTest form.
- Open Form1 (PortSettingsTest) and add the following code (in C#,):

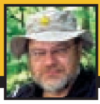
```
public partial class Form1 : Form
{
    private string portname = "Not Initialized";
    private string baudrate = "Not Initialized";
    private string parity = "Not Initialized";
    private string stopbits = "Not Initialized";
    private string databits = "Not Initialized";
    private string handshaking = "Not
    Initialized";

    // Instantiate the PortSettings class
    PortSettings p = new PortSettings();

    public Form1()
    {
        InitializeComponent();
    }

    private void clearSettings()
    {
        portname = "";
        baudrate = "";
        databits = "";
        stopbits = "";
        parity = "";
        handshaking = "";
    }

    private void buttonTest_Click(object sender,
    EventArgs e)
    {
        if (p.ShowDialog() ==
        DialogResult.Cancel)
        {
            // clear out settings
            clearLabels();
        }
        else
        {
            // set labels to Settings values
            setLabels();
        }
    }

    private void clearLabels()
    {
        labelPortName.Text = "PortName";
        labelBaudRate.Text = "BaudRate";
        labelParity.Text = "Parity";
        labelDataBits.Text = "DataBits";
        labelStopBits.Text = "StopBits";
        labelHandShaking.Text = "HandShaking";
    }

    private void setLabels()
    {
        labelPortName.Text = "PortName: " +
        p.selectedPort;
        labelBaudRate.Text = "BaudRate: " +
        p.selectedBaudrate.ToString();
        labelParity.Text = "Parity: " +
        p.selectedParity.ToString();
        labelDataBits.Text = "DataBits: " +
        p.selectedDataBits.ToString();
        labelStopBits.Text = "StopBits: " +
        p.selectedStopBits.ToString();
        labelHandShaking.Text = "HandShaking:
        " + wrap
          p.selectedHandshaking.ToString();
    }
}
```

- In the IDE, click the Debug menu and select the 'start debugging' item. You should see the blank PortSettingsTest form shown in **Figure 1**.
- Click the Test button and you should see the Settings form as shown in **Figure 6**.
- The actual port names will, of course (barring a stunning coincidence), be different from those shown. By highlighting and selecting these items and clicking Okay, the settings will propagate back to the PortSettingTest form as shown in **Figure 7**.

FIGURE 6. Settings Test.



We have covered a lot so far. In the last



You can find the source code and supplements for this article in Workshop19.zip at www.nutsvolts.com.

FIGURE 7. Settings Test Result.

Workshop, we created a GUI for our Simple Terminal. Then in this workshop, we created and tested a Settings form. You may be a bit tired of PC GUIs by now, but I have some good news. We will be able to use the Settings object in future projects by simply copying the source code to the new projects directory and using the Add/Existing Item technique shown here to get this object into our code. So, the good news is: *We never have to code this again*! Now you see one of the values of object-oriented programming. You can build and test a useful software object, forget all about how it works, and just use it like a black box.

The Serial Port Class

You may have looked at some of the uses of the Serial Port Class in the source code for the Simple Terminal from Workshop 18. You may have even had one of those 'what the ...' moments with some of the delegate stuff. Put simply, the code that is communicating with the serial port isn't actually running with the code for the GUIs. This concerns 'threads' or bits of software that are running separately in Windows (each gets a part of each second to run so that all the threads seem like they are running at the same time to the user). When you want one thread to put stuff in another thread, you set a delegate that the first thread can call from the second thread. So, we have to create a delegate in our GUI that allows the serial port thread to set the text in our richTextBox. Yeah, it's complex, but we don't really have to go too deep at this point just to use the serial port. This is one of those cases where cookbook coding really comes in handy; you can just safely use the source code and move on until you have time to burn out a few more brain cells and learn the details. If you get to that point, you might consider the *Cookbook* I mentioned earlier.

Now that we have built and tested the portSettings class, let's add it to the GUI that we built last month and get our Simple Terminal communicating between the PC and the Arduino.

Bringing It All Together

In Workshop 18, we built the Simple Terminal GUI. Now we want to add the portSettings stuff we just built so that we can make the GUI communicate using the serial port.

- Create a new Simple Terminal directory for your work and add these files (in C#) from the PortSetTest C# directory copy:
  - Port Settings.cs
  - Port Settings.Designer.cs
  - Port Settings.resz
  - DevInfo.cs
- Paste the files into your new directory.
- Following the Add/Existing Items instructions given earlier, add Port Settings and DevInfo to the project.
- In C# in the 'using' list, add:

```
using PortSet;
```

Now that you have added the PortSettings and DevInfo Classes to your Simple Terminal project, you are ready to use the .NET System.IO.Ports library that contains the SerialPort Class.

- Open the Form1 designer and from the Toolbox, drag and drop serialPort1.
- We will set the serial port properties according to the data we enter in the Port Settings class.
- In C#, add:

```
private void
settingsToolStripMenuItem_Click(object sender,
EventArgs e)
{
     // Make sure the port isn't already open
     if (serialPort1.IsOpen)
     {
          MessageBox.Show("The port must be closed
          before changing the settings.");
          return;
     }
     else
     {
          // creat an instance of the settings
          // form
          PortSettings settings = new
          PortSettings();

          if (settings.ShowDialog() ==
          DialogResult.OK)
          {
               if (settings.selectedPort != "")
               {
                    // set the serial port to the
                    // new settings
                    serialPort1.PortName =
                    settings.selectedPort;
                    serialPort1.BaudRate =
                    settings.selectedBaudrate;
                    serialPort1.DataBits =
                    settings.selectedDataBits;
                    serialPort1.Parity =
                    settings.selectedParity;
                    serialPort1.StopBits =
                    settings.selectedStopBits;

                    // show the new settings in the
                    // form text line
                    showSettings();
               }
               else
               {
                   MessageBox.Show("Error:
                   Settings form returned with wrap
                   no Serial port selected.");
                    return; // bail out
               }

          }
          else
          {
               MessageBox.Show("Error:
               buttonSetup_Click - Settings wrap
dialog box did not return Okay.");
               return; // bail out
          }

          // open the port
          try
          {
               serialPort1.Close();
               serialPort1.Open();
               menuStrip1.Items[1].Text =
               "Close Port";

               showSettings();
```

```
        }
        catch (System.Exception ex)
        {
            MessageBox.Show("Error -
            setupToolStripMenuItem_Click wrap
Exception: " + ex);
        }
    }
}

// show the settings in the form text line
private void showSettings()
{
    this.Text = "Smiley Micros - " +
        serialPort1.PortName + " " +
        serialPort1.BaudRate.ToString() + "," +
        serialPort1.Parity + "," +
        serialPort1.DataBits.ToString() + "," +
        serialPort1.StopBits;
    if (serialPort1.IsOpen)
    {
        this.Text += " - Port is open";
    }
    else
    {
        this.Text += " - Port is closed";
    }
}
```

- In the form designer, click on the 'Open Port' menu item.
- In C#, add:

```
private void
openPortToolStripMenuItem_Click(object sender,
EventArgs e)
{
    try
    {
        if (serialPort1.IsOpen)
        {
            serialPort1.Close();
            openPortToolStripMenuItem.Text
            = "Open Port";
        }
        else
        {
            serialPort1.Open();
            openPortToolStripMenuItem.Text =
            "Close Port";
        }

        showSettings();
    }
    catch (System.Exception ex)
    {
        MessageBox.Show("Error -
openPortToolStripMenuItem_Click Exception: wrap

              " + ex);
    }
}
```

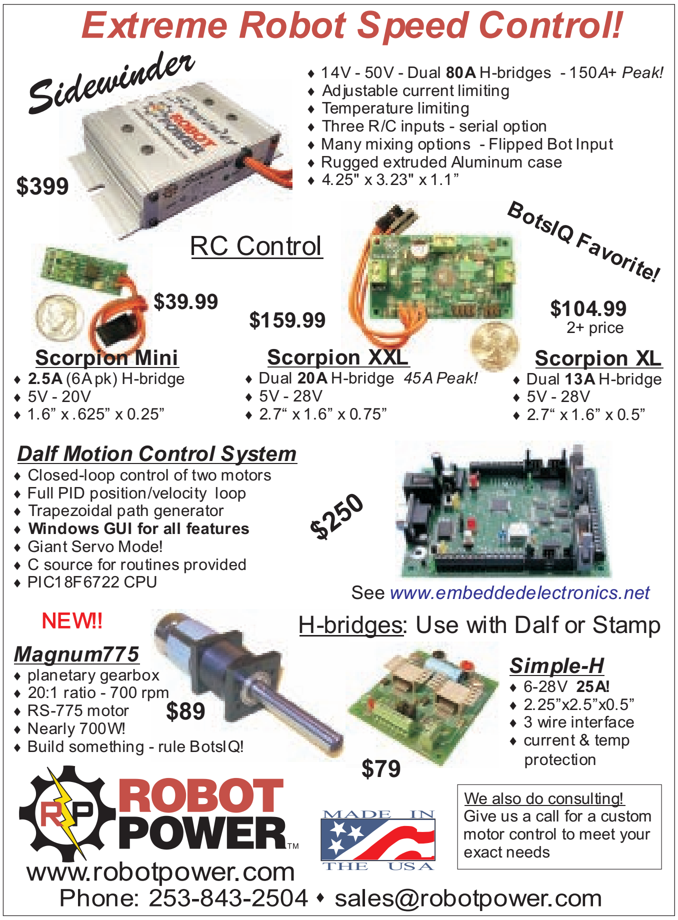
- In the form designer, select the richTextBoxSend component.
- In the properties window, click the 'KeyPress' event and to the event handler created in C#, add:

```
private void richTextBox1_KeyPress(object
sender, KeyPressEventArgs e)
{
sendChar(e.KeyChar);
}

private void sendChar(char c)
{
    char[] data = new Char[1];
    data[0] = c;
    try
    {
        serialPort1.Write(data, 0, 1);
    }
    catch
    {
        MessageBox.Show("Error: sendByte -
failed to send.\nIs the port open?");
    }
}
```

- Our receive functions use a delegate to allow the serial port read thread to write to our receive text box. This is a bit complex, so for the time being just use it as-is.
- In C#, add:

```
#region receive functions

// We want to have the serial port thread report
back data received, but to
// display that data we must create a delegate
// function to show the data in the
// richTextBox

// define the delegate
public delegate void SetText();
// define an instance of the delegate
SetText setText;

// create a string that will be loaded with the
// data received from the port
public string str = "";

// note that this function runs in a separate
// thread and thus we must use a
// delegate in order to display the results in
// the richTextBox.
private void serialPort1_DataReceived(object
sender,
System.IO.Ports.SerialDataReceivedEventArgs e)
{
    // instantiate the delegate to be invoked
    // by this thread
    setText = new SetText(mySetText);

    // load the data into the string
    try
    {
        str = serialPort1.ReadExisting();
    }
    catch (System.Exception ex)
    {
        MessageBox.Show("Error -
        port_DataReceived Exception: " + ex);
    }

    // invoke the delegate in the MainForm
    // thread
    this.Invoke(setText);
}

// create the instance of the delegate to be
// used to write the received data to
// the richTextBox
public void mySetText()
{
    // show the text
    richTextBox2.Text += str.ToString();

    moveCaretToEnd();
}

// This rigaramole is needed to keep the last
// received item displayed
private void
richTextBoxReceive_TextChanged(object sender,
System.EventArgs e)
{
    moveCaretToEnd();
}

private void moveCaretToEnd()
{
    richTextBox1.SelectionStart =
    richTextBox1.Text.Length;
    richTextBox1.SelectionLength = 0;
    richTextBox1.ScrollToCaret();
}
#endregion
```

So that will give you a working terminal program. Next month, we will use this code to create an Arduino voltmeter that shows ADC data on the PC monitor. **NV**


Extreme Robot Speed Control!
Sidewinder
$399
♦ 14V - 50V - Dual 80A H-bridges - 150A+ Peak!
♦ Adjustable current limiting
♦ Temperature limiting
♦ Three R/C inputs - serial option
♦ Many mixing options - Flipped Bot Input
♦ Rugged extruded Aluminum case
♦ 4.25" x 3.23" x 1.1"
RC Control
BotsIQ Favorite!
$39.99
$159.99
$104.99
2+ price
Scorpion Mini
♦ 2.5A (6A pk) H-bridge
♦ 5V - 20V
♦ 1.6" x .625" x 0.25"
Scorpion XXL
♦ Dual 20A H-bridge 45A Peak!
♦ 5V - 28V
♦ 2.7" x 1.6" x 0.75"
Scorpion XL
♦ Dual 13A H-bridge
♦ 5V - 28V
♦ 2.7" x 1.6" x 0.5"
Dalf Motion Control System
♦ Closed-loop control of two motors
♦ Full PID position/velocity loop
♦ Trapezoidal path generator
♦ Windows GUI for all features
♦ Giant Servo Mode!
♦ C source for routines provided
♦ PIC18F6722 CPU
$250
See www.embeddedelectronics.net
NEW!!
H-bridges: Use with Dalf or Stamp
Magnum775
♦ planetary gearbox
♦ 20:1 ratio - 700 rpm
♦ RS-775 motor
♦ Nearly 700W!
♦ Build something - rule BotsIQ!
$89
Simple-H
♦ 6-28V 25A!
♦ 2.25"x2.5"x0.5"
♦ 3 wire interface
♦ current & temp protection
$79
ROBOT POWER
MADE IN THE USA
We also do consulting!
Give us a call for a custom motor control to meet your exact needs
www.robotpower.com
Phone: 253-843-2504 ♦ sales@robotpower.com


WIRELESS MADE SIMPLE®
QUICKLY AND COST-EFFECTIVELY MAKE YOUR PRODUCT WIRELESS
RF MODULES
LOW COST • LONG RANGE
TRANSMITTERS • RECEIVERS • TRANSCEIVERS
INTEGRATED CIRCUITS
ENCODERS • DECODERS • TRANSCODERS
COMMAND • CONTROL • KEYLESS ENTRY
OEM PRODUCTS
EASILY CUSTOMIZED
ANTENNAS
FROM WHIPS TO CHIPS
Featured Products:
GPS Modules
GPS MADE SIMPLE™
RXM-GPS-SR
RXM-GPS-SG
• High Sensitivity • High Performance
• Easy Serial Interface • Ultra Compact
• Low Power Consumption • Low Cost
These easily applied GPS modules blend performance, and cost effectiveness into a single, compact, SMD package. The low-power chipset minimizes power consumption and provides exceptional sensitivity. Choose either an integrated or external antenna.
Development Systems
THE FAST TRACK TO WIRELESS SUCCESS
Linx development systems contain everything needed to evaluate the Linx product of you choice and implement it in record time.
LINX TECHNOLOGIES
800-736-6677
159 Ort Lane
Merlin, OR 97532
linxtechnologies.com


Complete Fabrication Center
AS 9100 Registration In Process
Quality prototype parts within 24 hours!!
Precision Laser, Waterjet, Plasma, Machining,
Micro-Machining, Forming, and Welding Capabilities
Parts from practically any material and from 0.001" to 6.000" in thickness.
Finishes such as powder coat, paint, plating, anodizing, silk screen, and more!
OUTPUTS
ANT 1 TO RF SWITCH
HF RCVR
RCDR
SW1
SW2
SW3
VHF/UHF RCVR
SA
SW4
SW5
SW6
Fabricated, silkscreened and shipped in 2 business days with no expedite charges!
* Quick
* Affordable
* Precise
* No Minimums
24 hour turn time based on quantity and finish requirements
Integrated Ideas & Technologies, Inc.
6164 W. Seltice Way • Post Falls, ID • 83854 • USA
Ph (208) 262-7200 • Fax (208) 262-7177 •
www.iitmetalfab.com

ALL ELECTRONICS CORPORATION

QUALITY Parts
FAST Shipping
DISCOUNT Pricing

CALL, WRITE, FAX or E-MAIL for a FREE 96 page catalog.
Outside the U.S.A. send $3.00 postage.

DUAL SPST SOLID STATE RELAY

Clare OptoMOS(R)#LAA110. Dual 1-Form-A Solid State Relay that has two independently controlled optically coupled MOSFET switches. 3750 Vrms of input to output isolation. Input: 1.2Vdc. Load current: 120mA. TTL/CMOS compatible. 8-pin DIP package. UL, CSA.
CAT# SRLY-110
$1.50 each

ULTRABRIGHT WHITE LED

Luxeon(R) III Star #LXHL-LW3C Compact, energy efficient, ultra-bright white LED. Rated for up to 1400mA operation. This Lambertian pattern emitter creates a cool, fully dimmable, 5500K white light. Operates on 3.7 - 3.9 Vdc. Aluminum-backed module is 20mm dia. x 7.2mm. **CAT# LED-144**
$5.00 each
10 for $4.75 each

FERRITE SNAP-BEAD

TDK #ZCAT1518-0730-BK. For cables up to 7mm (0.27") dia. Case has an eyelet on one end which can be used with a wire-tie for extra fastening security. Case size 15mm dia. x 18mm long. Provides excellent absorption of high-frequency EMC. **CAT# FB-85**
$1.00 each
100 for 65¢ each
900 for 45¢ each

24VDC 6.5A 150W POWER SUPPLY

Mean Well #RS-150-24. 197 x 98 x 38mm. Short circuit/ overload / over-voltage protection. Adjustable output voltage. Led output indicator. UL, CSE, TUV, CE.
CAT# PS-24150
$26.95 each
20 for $25.95 each · 100 for $24.95 each

24VDC 350W MOTOR

High-performance 24 Vdc motor for electric scooters and bicycles, Schwinn, GT, Mongoose, Izip (Currie Technologies). Standard on other electrics as well. 350 watts. 2600 RPM. 22 Amps. 4" diameter x approximately 4" long. Three-hole mounting flange - 0.25" diameter holes on approx. 3.9"centers. 3/8" (8mm) dia. outer shaft, 10mm dia. inner shaft, equipped with an 11 tooth sprocket. 24" long lead.
CAT# DCM-1353
$20.00 each

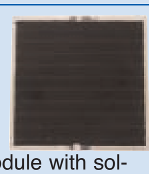
SPDT KEY SWITCH

S.P.D.T., on-on, keyswitch. Key removable in both positions. Rated 3A 250V. 19mm threaded bushing mount. Good-looking chrome bezel. Nice, snappy feeling when switching from one position to the other. All switches are keyed alike. Includes one key.
CAT# KSW-29
$5.75 each

10-CONDUCTOR, 3" JUMPER

3" ribbon jumper cable. 10 leads, stripped and tinned, on 0.1" centers. RoHS compliant. **CAT# RJ-310**
50¢ each
100 for 30¢ each
500 for 25¢ each

SOLAR CELL



Output: approximately 3 Volts @ 40 mA. 60mm square x 2.5mm thick epoxy-encapsulated silicon photovoltaic cell. Solid, almost-unbreakable module with solderable foil strips on backside. Ideal for solar-powered battery chargers and other projects.
CAT# SPL-61
$3.75 each
100 for $3.25 each

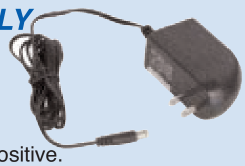
LARGE, SURFACE-MOUNT PUSHBUTTON

S.P.S.T., normally-open, momentary pushbutton switch. 1.6" diameter metal body with mounting flanges, holes on 2.3" centers. Originally for automotive horn application. 0.84" diameter black push-button, 0.95" overall height. Packaged with 2 mounting screws.
CAT# PB-156
$1.75 each

ENCASED PIEZO ELEMENT, PC MOUNT

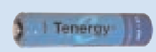
Murata #PKM22EPP-40. 22mm diameter x 8mm plastic case. PC pins on 10mm centers. 4KHz resonant frequency. Piezo element inside case is 20mm diameter.
CAT# PE-57
75¢ each

12 VDC 1.5 A SWITCHING POWER SUPPLY



Input: 100-120Vac, 50/60Hz. Output: 12Vdc 1.5A. 6' cord. 2.1mm coax power plug. Center positive. UL. **CAT# PS-12151**
$9.50 each
10 for $9.00 each

RECHARGEABLE NICKEL METAL HYDRIDE BATTERIES



AAA CELL 1.2 Volt, 1000 mAh. 0.4" dia. X 1.74" long.
CAT# NMH-AAA $2.50 each
10 for $2.10 each

AA CELL, 2600 mAH
ULTRA-HIGH CAPACITY
1.2 Volt, 2600 mAH. 0.56" dia. x 1.99" long.
CAT# NMH-AA $2.95 each
8 for $2.75 each • 60 for $2.40 each

Shop ON-LINE www.allelectronics.com
ORDER TOLL FREE 1-800-826-5432

MAIL ORDERS TO:
ALL ELECTRONICS CORP.
14928 OXNARD ST., VAN NUYS, CA 91411-2610

FAX (818) 781-2653 · INFO (818) 904-0524
E-MAIL allcorp@allcorp.com

NO MINIMUM ORDER • All Orders Can Be Charged to Visa, Mastercard, American Express or Discover • Checks and Money Orders Accepted by Mail • Orders Delivered in the State of California must include California State Sales Tax • NO C.O.D • Shipping and Handling $7.00 for the 48 Continental United States - ALL OTHERS including Alaska, Hawaii, P.R. and Canada Must Pay Full Shipping • Quantities Limited • Prices Subject to change without notice.

MANUFACTURERS - We Purchase EXCESS INVENTORIES... Call, Write, E-MAIL or Fax YOUR LIST.

PICAXE PRIMER

SHARPENING YOUR TOOLS OF CREATIVITY

■ BY RON HACKETT

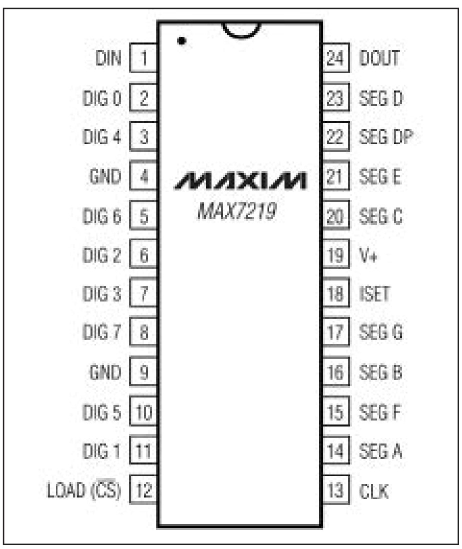
USING THE MAX7219 LED DISPLAY DRIVER

In the previous installment of the PICAXE Primer, we experimented with interfacing a seven-segment LED display with our master processor. This month — in order to simplify the circuitry and software normally required to drive multiple-digit seven-segment LED displays — we're going to explore the capabilities of the MAX7219 eight-Digit LED display driver. Once we have covered the basics of interfacing with the 7219, we'll use it to develop a serially interfaced four-digit peripheral LED display that can be used in any PICAXE project — even one powered by the little 08M.

THE MAX7219 EIGHT-DIGIT LED DISPLAY DRIVER

The MAX7219 LED display driver is capable of interfacing a microprocessor with as many as eight common-cathode seven-segment LED displays, bar-graph displays up to 64 bars, or a total of 64 individual LEDs. Among its many features, the following will be especially helpful as we develop our four-digit peripheral LED display:

■ FIGURE 1. MAX7219 Pinout.



- An internal "scan-limit" register allows the user to display from one to eight digits.
- Digits may be updated individually without having to update the entire display.
- Only three I/O pins are required for the interface.
- Internal BCD (Binary Coded Decimal) decoding eliminates the need for lookup tables.
- All multiplexing circuitry is internal which greatly simplifies programming requirements.
- One external resistor is all that is required to set the segment current for all LED segments.
- A simple digital brightness control can be set via software.

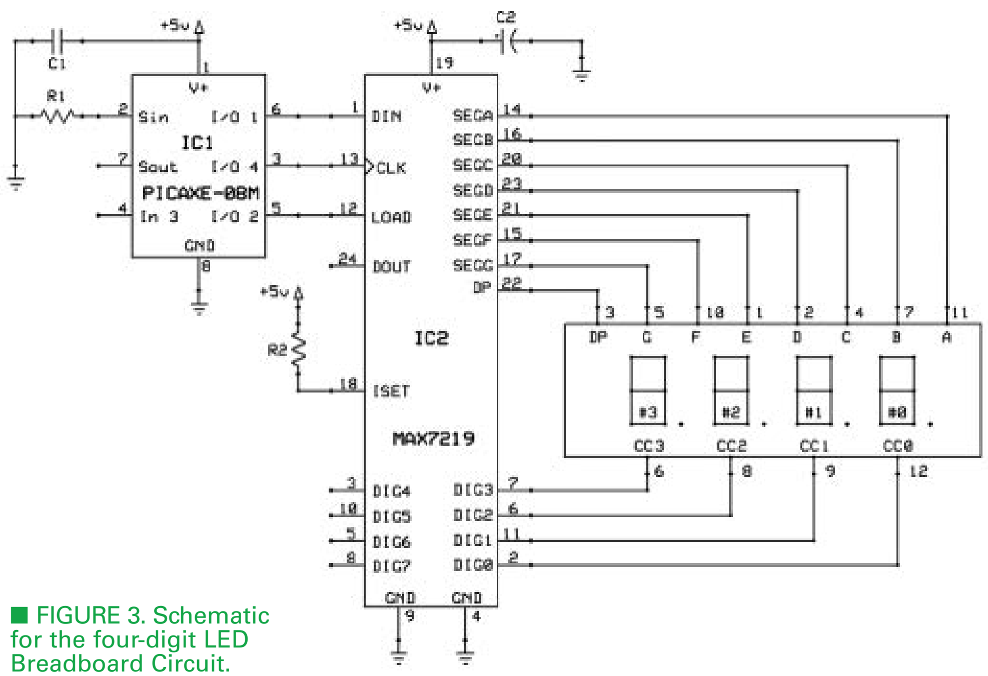
Figure 1 presents the pinout of the MAX7219 and **Figure 2** lists the functions of each of its pins. The eight "segment" pins and the eight "digit" pins are very straightforward; each of these 16 outputs directly connects to the corresponding pin of the LED display. If you are using fewer than eight LED digits (as we are), it's important to connect the outputs with the lowest numbers first because the 7219's internal scan-limit register works from the bottom up. In other words, you can limit the automatic multiplexing to digits 0-3, but not to digits 4-7.

As indicated in **Figure 2**, the ISET pin controls the current that flows through the LED segments.

■ FIGURE 2. MAX7219 Pinout Functions.

Pin	Name	Function
19, 4, 9	V+ & Gnd	Power & Ground: Both Ground pins must be connected.
2, 3, 5-8, 10, 11	DIG 0 - DIG 7	Connect to the corresponding Digit inputs on the LED.
14-17, 20-23	SEG A - G & DP	Connect to the corresponding Segment inputs on the LED.
18	ISET	Connect through a current-limiting resistor to V+
1	DIN	Serial data-in: A 16-bit data word is shifted in on this pin.
24	DOUT	Data-out pin: Only used when cascading multiple 7219s.
13	CLK	Serial data input clock: Each bit shifted in on rising edge.
12	LOAD	Load data input: Pulse this pin to store 16-bit input word

According to the MAX7219 datasheet (available on the *Nuts & Volts* website), the current through each LED segment is approximately 100 times the current through the ISET resistor. Table 11 in the datasheet lists suggested resistor sizes for various segment currents. I chose to use a 47K resistor which results in more than adequate brightness for the LED display that I am using. You can certainly adjust that value up or down, but don't go below the recommended minimum of 10K. That value results in a segment current of about 40 mA or 320 mA when all eight segments of a digit are lighted. Since the 7219's maximum current-handling capability is listed as 330 mA, segment currents greater than 40 mA could damage or destroy the 7219. (The number of digits in the display doesn't matter because the display is multiplexed, so only one digit at a time is powered.)



■ FIGURE 3. Schematic for the four-digit LED Breadboard Circuit.

COMMUNICATING WITH THE MAX7219

At this point, we have discussed all the MAX7219's pins except for the three-pin interface to the microprocessor. **Figure 3** presents the schematic for our first experiment with the 7219; you may want to refer to it during the following discussion. In the **schematic**, you can see the three required connections between the 08M and the 7219. (The only significance to the specific interconnections that I chose is that they result in a simpler stripboard layout.)

For each data byte that we want to send to the 7219, we have to serially shift a 16-bit data word into the 7219's internal input register via the DIN pin. The most significant bit (MSB) of the word should be sent first; the high byte of the word should contain the address of the 7219 internal register into which we want to place the data, and the low byte should contain the value of the data.

Figure 4 is a listing of addresses for the MAX7219's 14 internal Digit and Control registers. The eight Digit registers simply hold the value that we want to display on each corresponding LED digit. The six Control registers require a brief explanation:

No-Op (address 0): This register is used when cascading MAX7219s to drive more than eight LED digits. We won't be doing that, but you can consult the datasheet for details.

Decode Mode (address 9): The value placed in this register determines whether BCD decoding will be used for each of the display digits. (We'll discuss BCD decoding shortly.)

Intensity (address 10): This register can contain a value between 0 and 15 (inclusive) which determines the brightness of the LED display (0 = minimum; 15 = maximum). This setting is in addition to that of the ISET resistor. In other words, the display can be dimmed or brightened under program control.

Scan Limit (address 11): The value placed in this register corresponds to the maximum digit number that will be included in the automatic multiplexing of the display. For example, we'll be using digits 0 through 3, so we need to place a "3" in the Scan Limit register.

Shutdown (address 12): This register controls whether the display is turned on or off (1 = on; 0 = off).

Display Test (address 15): This register can be used to test the display by lighting all the LED digits at once (0 = normal operation; 1 = test mode). We won't be using this register.

Before we go any further, let's discuss the 7219's BCD decoding capability. In the previous Primer installment, we needed to develop a

■ FIGURE 4. MAX7219 Internal Digit and Control Registers.

REGISTER	ADDRESS					HEX CODE
	D15–D12	D11	D10	D9	D8	
No-Op	X	0	0	0	0	0xX0
Digit 0	X	0	0	0	1	0xX1
Digit 1	X	0	0	1	0	0xX2
Digit 2	X	0	0	1	1	0xX3
Digit 3	X	0	1	0	0	0xX4
Digit 4	X	0	1	0	1	0xX5
Digit 5	X	0	1	1	0	0xX6
Digit 6	X	0	1	1	1	0xX7
Digit 7	X	1	0	0	0	0xX8
Decode Mode	X	1	0	0	1	0xX9
Intensity	X	1	0	1	0	0xXA
Scan Limit	X	1	0	1	1	0xXB
Shutdown	X	1	1	0	0	0xXC
Display Test	X	1	1	1	1	0xXF

7-SEGMENT CHARACTER	REGISTER DATA						ON SEGMENTS = 1							
	D7*	D6–D4	D3	D2	D1	D0	DP*	A	B	C	D	E	F	G
0		X	0	0	0	0		1	1	1	1	1	1	0
1		X	0	0	0	1		0	1	1	0	0	0	0
2		X	0	0	1	0		1	1	0	1	1	0	1
3		X	0	0	1	1		1	1	1	1	0	0	1
4		X	0	1	0	0		0	1	1	0	0	1	1
5		X	0	1	0	1		1	0	1	1	0	1	1
6		X	0	1	1	0		1	0	1	1	1	1	1
7		X	0	1	1	1		1	1	1	0	0	0	0
8		X	1	0	0	0		1	1	1	1	1	1	1
9		X	1	0	0	1		1	1	1	1	0	1	1
—		X	1	0	1	0		0	0	0	0	0	0	1
E		X	1	0	1	1		1	0	0	1	1	1	1
H		X	1	1	0	0		0	1	1	0	1	1	1
L		X	1	1	0	1		0	0	0	1	1	1	0
P		X	1	1	1	0		1	1	0	0	1	1	1
blank		X	1	1	1	1		0	0	0	0	0	0	0

*The decimal point is set by bit D7 = 1

■ FIGURE 5. MAX7219 BCD Decoding Chart.

lookup table in order to convert the value of a specific digit to the correct corresponding pattern of segments that needed to be lit to display the digit. The 7219 eliminates this programming chore by internally performing the necessary data conversions. **Figure 5** presents the 16 characters that the 7219 can automatically decode; these include the numerical digits 0-9 (as well as a blank space) and the alpha characters "H," "E," "L," and "P." (As you can see, the BCD values for the 10 digits conveniently correspond to the digit itself.) If you want to light the decimal point to the right of any character, bit7 of the corresponding data value must be set to "1." For example, the BCD value for "8" is 8 and the BCD value for "8" is 136 because bit7 of a binary number has a value of 128 — 128 + 8 = 136.

The 7219's decoding scheme is extremely flexible in that it allows us to individually enable or disable BCD decoding for each of the display's digits by setting or resetting the corresponding bits in the value stored in the Decode Mode register. We're going to take the simplest approach for our first experiment and decode all four of our display's digits, so we need to store %00001111 (decimal 15) in the Decode Mode register. If you want to display characters that have values other than the 16 BCD values shown in **Figure 5**, you will need to turn off BCD decoding for whichever digits you want to use and implement a lookup table in software to retrieve the necessary values — consult the 7219's datasheet for details.

■ FIGURE 6. Parts List for the four-digit LED Breadboard Circuit.

ID	Part
R1	Resistor, 100k, 1/4 W
R2	Resistor, 47k, 1/4 W
C1	Capacitor, .01uF
C2	Capacitor, electrolytic, 10uF
IC1	PICAXE-08M
IC2	MAX7219 Display Driver
-	LED display, 7 segments, 4 digits

Now that we have discussed the details of the two bytes that make up the 16-bit data word that needs to be sent to the 7219, let's take a look at the process of serially sending the data. The sequence needs to proceed as follows:

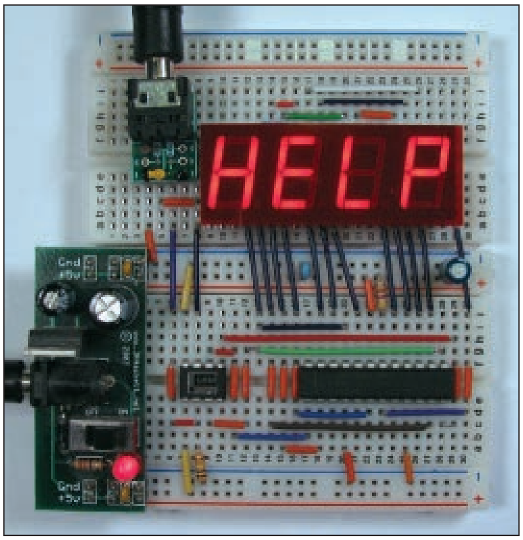
1. Beginning with the MSB of the 16-bit data word, a bit is loaded onto the 08M output connected to the 7219's DIN pin.
2. The output connected to the CLK pin is briefly pulsed to transfer the bit to the 7219's internal input register.
3. The data word is shifted one place to the left to move the next bit into the MSB position. The easiest way to do this is to simply multiply the data word by two. (Try a few examples with binary numbers and you will see why this works.)
4. The first three steps are repeated for each of the bits in the data word.
5. Once the entire 16-bit data word has been shifted into the 7219, the output connected to the LOAD pin is briefly pulsed to latch the data into the appropriate registers in the 7219.

If you are a regular PICAXE Primer reader, you know that more than once I have discussed the care that needs to be taken when declaring word variables in a PICAXE program. For example, if you declare a word variable for w0 (e.g., symbol outword = w0), I have warned that you need to be careful to avoid also declaring an additional variable for either b0 or b1 (e.g., symbol maxreg = b0 or symbol outbyte = b1) because changing the value of one variable (e.g., b1) will inadvertently also change the value of another variable (w0, in this case). However, as with most rules, there's always an exception and the present situation is one of those exceptions.

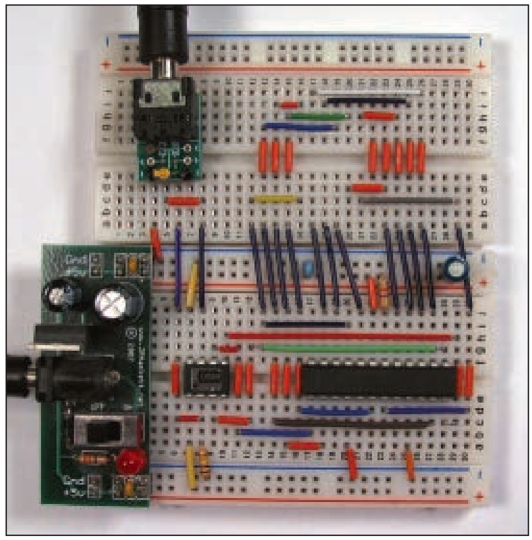
As you will see when we get to the software for our first experiment, we're going to make the exact three declarations that I just used as examples. The reason is simple: We need a 16-bit word variable (outword) that we can serially shift out to the 7219, and we also need two separate byte variables (maxreg and outbyte) that we can individually update as the need arises. In effect, outword is a concatenation of maxreg and outbyte – when we change the value of maxreg or outbyte for our next transmission to the 7219, we are also intentionally changing the value of outword which is the 16-bit variable we need to shift out to the 7219.

ASSEMBLING AND TESTING A MAX7219 BREADBOARD CIRCUIT TO DRIVE AN LED DISPLAY

To see how all this works, let's



■ FIGURE 7. Breadboard Circuit in Operation.



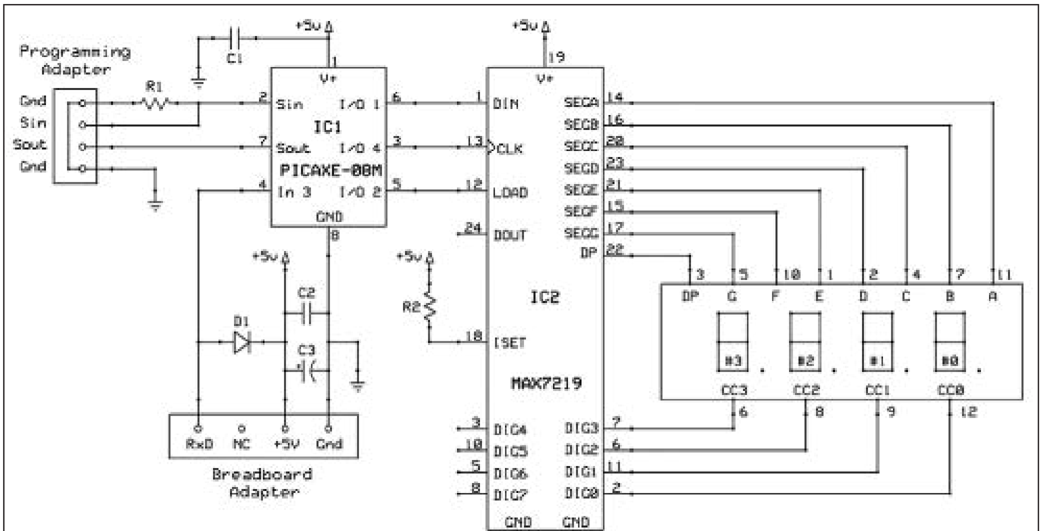
■ FIGURE 8. Breadboard Circuit with LED Display Removed.

assemble the breadboard circuit for our first experiment. The schematic was presented earlier in **Figure 3** and the parts list is contained in **Figure 6**. Similarly to our previous project, I haven't included the standard PICAXE programming circuitry in the schematic because it's included on the programming adapter that I'm using. **Figure 7** is a photo in the completed breadboard circuit for our first experiment. As you can see, it was a bit of a challenge to fit all the circuitry on the two small breadboards. If you would rather have more space to work with, you could certainly use larger breadboards. In the photo in **Figure 7**, I have placed a piece of red transparent plastic on top of the LED display because the lighted display photographs better that way. **Figure 8** is a photo of the same setup with the four-digit display removed so you can see the wiring underneath. The upper six pins of the display fit in the six vacant holes just above the mid-line of the breadboard between the vertical orange jumpers.

Once you have assembled your breadboard circuit, we're ready to move on to the software portion of our experiment. Our first program for the LED board will take full advantage of the MAX7219's automatic BCD decoding to simply print the message "HELP" on the display. The necessary software (LED7219help.bas) is available on the *N&V* website. Download it and take a few minutes to read through it to see how the details of our discussion of the 7219 have been implemented in the software. When you understand how the program functions, use the Programming Editor to install it to your breadboard circuit. If "HELP" doesn't immediately arrive, you will need to troubleshoot the wiring of your breadboard circuit.

DESIGNING A STRIPBOARD VERSION OF THE MAX7219 CIRCUIT

When your breadboard setup is functioning correctly, we're ready to construct the stripboard version of the project. **Figure 9** presents the schematic for the circuit, which is essentially the same as the breadboard version with the addition of input circuitry to receive the serial input from the master processor and the necessary headers to connect to the breadboard, LED, and programming adapter. There is, however, one addition to the circuit that requires a brief explanation. The input circuitry includes a 1N4148 diode (D1) that is connected between the 08M's data input pin (input 3) and +5V. As indicated in the documentation for the serin command in Part II of the PICAXE manual, this diode is necessary whenever the 08M's input 3 is used to receive serial data. Also, be sure to connect the cathode of the diode (not the anode) to +5V as



■ FIGURE 9. Schematic for the four-digit LED Stripboard Circuit.

ID	Part
-	Stripboard, 20 traces x 25 holes
-	Jumper wire
D1	Diode, 1N4148
R1	Resistor, 100k, 1/4 W
R2	Resistor, 47k, 1/4 W
Prog Adapter	Header, female, 4-pin rt. angle
BB Adapter	Header, female, 4-pin rt. angle
C1, C2	Capacitor, .01uF
IC1	DIP socket, 8 machine-pin
-	PICAXE-08M
IC2	DIP socket, 24 machine-pin
-	MAX7219 Display Driver
C3	Capacitor, electrolytic, 10uF
-	LED display, 4-digit, 7-segment
-	Header, male, 5-pin, double-ended (for connecting to Breadboard)
optional (see text)	2 Headers, male, 5-pin straight (Reverse-mountable)
optional (see text)	2 Headers, female, 6-pin straight (for LED display)

■ FIGURE 10. Parts List for the four-digit LED Stripboard Circuit.

shown in the **schematic**.

Figure 10 lists the necessary parts which are also very similar to those of the breadboard circuit we constructed earlier. The additional parts include the necessary headers and the 1N4148 diode we just discussed. The stripboard layout is shown in **Figure 11**. Before you actually begin assembling the circuit, there are a couple of aspects of the layout that I want to clarify.

First, as you can see, the stripboard needs to contain 20 traces with 25 holes each. By now, you probably know that the smaller stripboard that's available on my site has 19 traces with holes. However, it also has one trace on each side without any holes whatsoever, for a total of 21 traces. In the layout, all the traces are shown as having holes in them, but I actually used the small stripboard with one "hole-less" trace along its left edge (at column A).

There are two jumpers that need to terminate in the "column A" trace: one from S7 to A7, and one from E21 to A21. Rather than drilling two holes in trace A (at A7 and A21), you can see in the layout that I inserted one end of each of these jumpers in trace B and extended it on the bottom of the board so that I could solder it at trace A. (I also severed trace B appropriately so that I didn't make contact with any other circuit connection.) You can take the same approach or drill the necessary two holes. If you prefer, you could also use the larger stripboard so that you have holes in trace A, and insert one end of each of those two jumpers directly into trace A.

Second, there are four jumpers and one resistor that are shown in the layout as running on top of the two IC sockets. I did this so that the required connections would be clear in the layout. However, these five parts need to be installed underneath the sockets (which is why we are using machined-pin sockets); be sure to install these parts before soldering the sockets in place. Also, there is one more jumper (from D25 to P25) that is shown as running on top of another component (the breadboard connector).

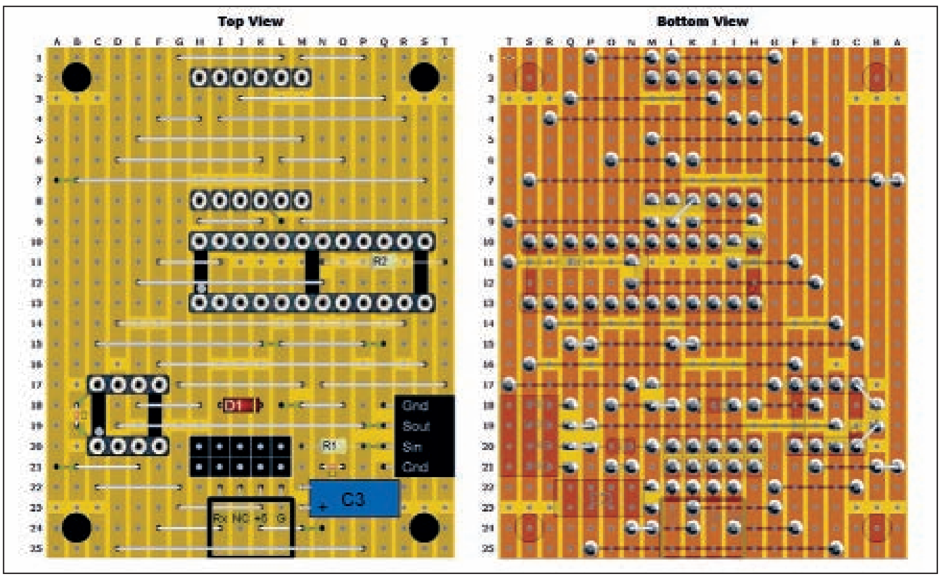
In this case, that's correct; in addition to carrying a signal (the sin connection), this jumper also serves the purpose of "pinning" the breadboard connector to the stripboard so that the connector doesn't bend and weaken as the result of repeated insertions and extractions into and out of the breadboard. You may also want to add an additional jumper over the top of the programming adapter (but I don't think that connector will be stressed near as much as the breadboard connector). If you do, be sure to insulate it to avoid any accidental connection to the jumper that terminates at T17.

Finally, the two six-pin straight female headers shown in rows 2 and 8 are optional; you can include them if you want to be able to remove the LED display from the board, or you can solder the LED directly into the same holes if you prefer. The two five-pin, reverse-mountable male headers just above the breadboard connector are also optional. If you want to be able to locate the LED display some distance from your master processor circuit, you can connect a 10-pin ribbon cable assembly from the bottom of the stripboard to the master processor circuit. You can also add these headers at a later time if you want to.

ASSEMBLING THE STRIPBOARD CIRCUIT

As usual, read through the complete list of assembly instructions that follows to be sure you understand the entire procedure before assembling the board. As you carry out the following steps, you may want to refer to the photo of the completed stripboard circuit

■ FIGURE 11. Layout for the four-digit LED Stripboard Circuit.



that is presented in **Figure 12**.

Cut and sand a piece of stripboard to the required size (20 traces with 25 holes each).

Sever the traces on the bottom of the board as indicated in **Figure 11**.

Clean the bottom of the board with a plastic "Scotch-Brite" or similar abrasive pad.

Insert all the jumpers, except the one from D25 to P25. Be sure to allow enough length for the jumpers that need to be extended on the bottom of the board.

Flip the board over, bend, and snip the leads that need to be extended to adjacent traces. Solder and snip all the leads, except the one that extends from P19 to Q19. (We'll solder that one when the header is installed.)

Observing the correct polarity, insert the diode; solder and snip its leads.

If you want to include the optional 5x2 reverse-mountable male headers for a ribbon cable connection, insert it and solder it in place at this point; snip the short ends from the top of the board and file them smooth.

Insert the two resistors. Flip the board over; bend and snip the lead that needs to be extended from P20 to Q20. Solder and snip all the leads, except the one that extends from P20 to Q20. (We'll also solder that one when the header is installed.)

Insert the four-pin, right-angle female header for the programming adapter. Flip the board over with the header in place and set it on a flat surface.

Make sure the appropriate leads are touching the pins at Q19 and Q20; solder the four pins and the two leads at P19 and P20.

Insert the capacitor C1 (but not C2); solder and snip its leads.

Insert the four-pin, right-angle female header for the breadboard connector. Flip the board over with the header in place, set it on a flat surface and solder the four pins.

Insert capacitor C2 and both machined-pin IC sockets, making sure that pin 1 is in the correct position for both sockets.

Flip the board over; bend and snip the two capacitor leads so that they make contact with pins 1 and 8 of the eight-pin IC socket. Solder the sockets in place and the capacitor leads at B18 and B19.

Use an insulated jumper for the one that runs from D25 to P25 and installs over the top of the breadboard connector. Install it and (using a small clamp to hold it and the header tightly against the stripboard) solder it in place and snip its leads.

Observing the correct polarity, insert capacitor C3 and bend its leads so that the capacitor lies flat against the board as shown in **Figure 12**; solder and snip its leads.

If you want to use two six-pin straight female headers to mount the LED display, install them now.

Flip the board over; bend and snip a short jumper around the header pin at K8 so that it extends to L9. Solder the headers in place, and the jumper at L9

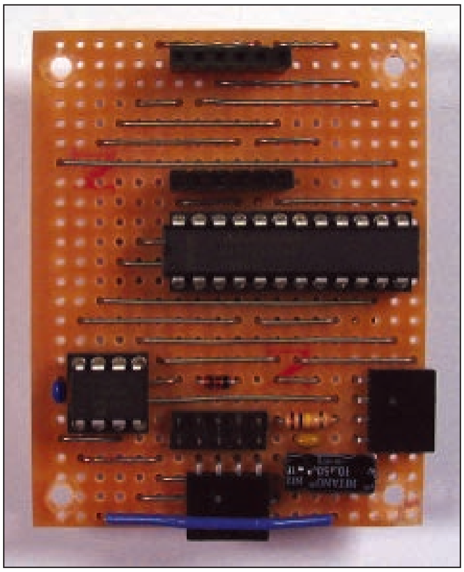
If you prefer to solder the LED display directly to the board, do so now by following the same procedure outlined in the previous step.

Clean the flux from the bottom of the board and allow it to dry.

Inspect the board carefully for accidental solder connections or other problems.

TESTING THE STRIPBOARD CIRCUIT

In order to test the completed stripboard circuit, you will need to insert the board into a breadboard and make the necessary power and ground connections (see **Figure 11**). For software, we can simply re-use



■ FIGURE 12. Top View of the Completed Stripboard Circuit.

the same program (LED7219help.bas) that we used to test our breadboard circuit. As you can also see in **Figure 11**, the pin-out of the programming connector is the same as we have used previously, so you should already have the necessary programming adapter. Just connect it to the display board and download LED7219help.bas; "HELP" should again arrive!

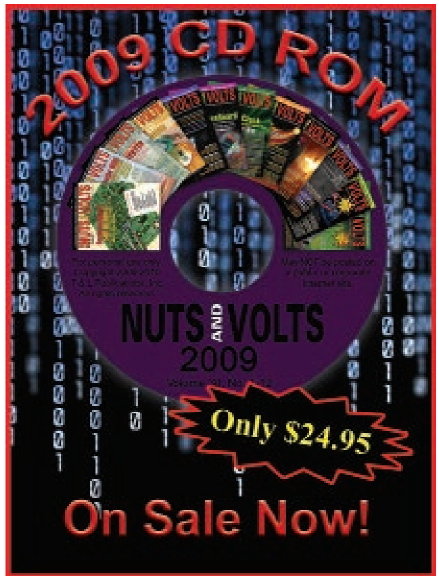
I was sure that we would finish our LED display project this time but, as usual, I talk too much. In the next installment of the Primer, we'll experiment with two different programs for our display. First, we'll develop a simple program to count from 0 to 9999, and we'll throw in a little "zero-blanking" for good measure. Then, we'll install driver software on the LED display that will enable our master processor to send serial data to it using only one output pin. While you're waiting for April to arrive, you may want to see what you can come up with on your own. See you then ... **NV**

Check the *Nuts & Volts* website at **www.nutsvolts.com** for accompanying downloads. Check Ron's website at **www.jrhackett.net** for items from the Parts Lists.

We have over 180 related titles in our webstore!!

The Nuts & Volts WEBSTORE

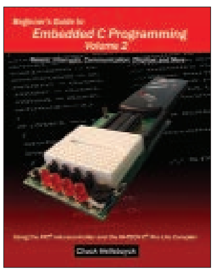
CD-ROMS SPECIALS



ELECTRONICS

Beginner's Guide to Embedded C Programming Volume 2
by Chuck Hellebuyck

In this "Volume 2," Chuck takes the reader to the next level by introducing how to drive displays, how to use interrupts, how to use serial communication, and how to use the internal hardware peripherals of the PIC16F690 microcontroller such as SPI, PWM, and timers.



When you have finished reading this book and completed the projects, you'll be well beyond the title of Beginner!

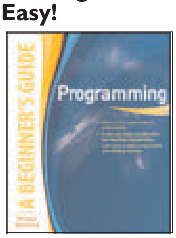
$39.95

The components that are added to the PICKit 2 that Chuck uses in his new book can also be purchased through the *Nuts & Volts* Webstore.

$12.95

Programming A Beginner's Guide
by Richard Mansfield

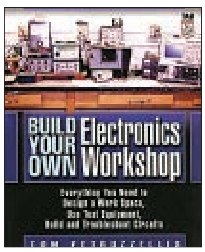
Essential Programming Skills--Made Easy!



This book will get you started right away writing a simple but useful program in Visual Basic Express Edition, and then moves on to more advanced projects, including a quiz program and a protected personal diary. You'll develop real-world programming skills, like designing user interfaces and working with variables, arrays, loops, and procedures. **$29.95**

Build Your Own Electronics Workshop
by Thomas Petruzzellis

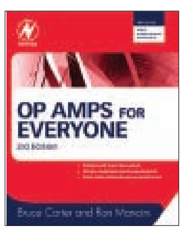
YOUR DREAM ELECTRONICS LAB IS WAITING INSIDE!



This value-packed resource provides everything needed to put together a fully functioning home electronics workshop! From finding space to stocking it with components to putting the shop into action — building, testing, and troubleshooting systems — popular electronics author Tom Petruzzellis' *Build Your Own Electronics Workshop* has it all! And the best part is, this book will save you money, big time!

Reg $29.95 Sale Price $24.95

Op Amps for Everyone, Third Edition
by Bruce Carter and Ron Mancini

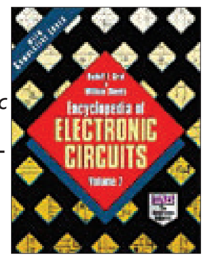


OP AMPS FOR EVERYONE provides the theoretical tools and practical know-how to get the most from these versatile devices - this new edition substantially updates coverage for low-speed and high-speed applications, and provides step by step walkthroughs for design and selection of op amps and circuits.

$79.95*

Encyclopedia of Electronic Circuits, Volume 7
by Rudolf F. Graf / William Sheets

NEW IN THE BESTSELLING SERIES! ONE THOUSAND MORE LEADING-EDGE CIRCUIT DESIGNS!

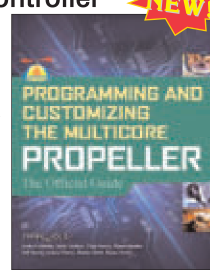


Designed for quick reference and on-the-job use, the *Encyclopedia of Electronic Circuits, Volume 7*, puts over 1,000 state-of-the-art electronic and integrated circuit designs at your fingertips. Organized alphabetically by circuit type, this all-new collection includes the latest designs from industry giants such as Advanced Micro Devices, Motorola, Teledyne, General Electric, and others.

$39.95

Programming and Customizing the Multicore Propeller Microcontroller
by Parallax

NEW!



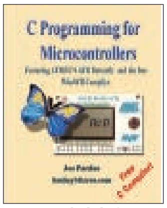
Written by a team of Propeller experts, this authoritative guide shows you how to realize your design concepts by taking full advantage of the multicore Propeller microcontroller's unique architecture. The book begins with a review of the Propeller hardware, software, and Spin language so you can get started right away.

$49.95*

Order online @ **www.nutsvolts.com**

Or CALL **1-800-783-4624** today!

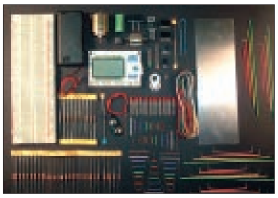
BOOK & KIT COMBOS



From the Smiley Workshop

C Programming for Microcontrollers

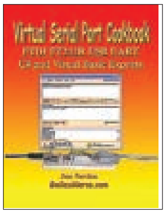
by Joe Pardue



Book **$44.95** Kit **$66.95**

Do you want a low cost way to learn C programming for microcontrollers? This 300 page book and software CD show you how to use ATMEL's AVR Butterfly board and the FREE WinAVR C compiler to make a veryinexpensive system for using C to develop microcontroller projects.

Combo Price $99.95 Plus S/H



Virtual Serial Port Cookbook

by Joe Pardue

As talked about in the *Nuts & Volts* June issue, "Long Live The Serial Port"

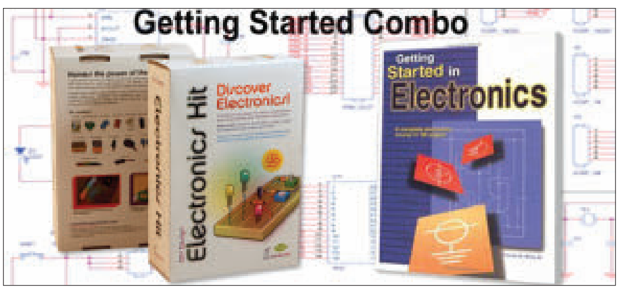
Book **$44.95** Kit **$69.95**

This is a cookbook for communicating between a PC and a microcontroller using the FTDI FT232R USB UART IC. The book has lots of software and hardware examples. The code is in C# and Visual Basic Express allowing you to build graphical user interfaces and add serial port functions to create communications programs.

The Virtual Serial Port Parts Kit and CD

(also *available, above right*)

Reg. Price $ 114.90 **Subscriber Price $109.95 Plus S/H**



The Getting Started Combo includes: Getting Started in Electronics by Author Forrest Mims and the DIY Electronics Kit. In his book, Mims teaches you the basics and takes you on a tour of analog and digital components. He explains how they work and shows you how they can be combined for various applications. The DIY Electronics Kit allows for the hands-on experience of putting circuits together -- the kit has over 130 parts! No soldering is required and it includes its own 32 page illustrated manual. **Combo Price $62.95 Plus S/H**

THE EVIL GENIUS SERIES

If you like projects, The Evil Genius Series is for you!

We have all 26 in stock and ready to ship.

Look for special offers.

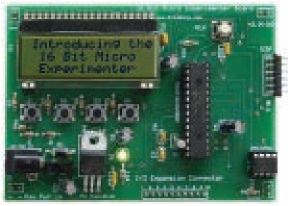
http://store.nutsvolts.com

CD-ROMS

CALL 1-800-783-4624 today!
Or Order online @ www.nutsvolts.com

PROJECTS

16 Bit Micro Experimenter Board



Ready to move on from eight-bit to 16-bit microcontrollers? Well, you're in luck! In the December 2009 Nuts & Volts isssue, we will introduce you to a new, "the 16-bit Micro Experimenter". The kit comes with a CD-ROM that contains details on assembly, operation, as well as an assortment of ready-made applications. New applications will be added in comming months.

Subscriber's Price **$55.95**
Non-Subscriber's Price **$59.95**

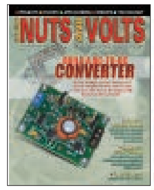
3-Phase Wind Turbine Kit

NEW!

The Whirlybird 3-Phase Wind Turbine Kit is one of the great projects from the series of articles by John Gavlik, "Experimenting with Alternative Energy." In Parts 6 and 7, he teaches you how to produce the most electricity utilizing the wind.
For kit details, please visit our webstore.

Subscriber's Price **$75.95**
Non-Subscriber's Price **$79.95**

DC-to-DC Converter Kit



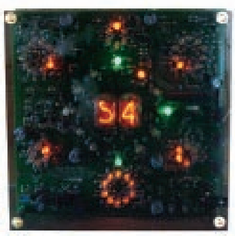
As seen on the March 2009 cover

"Roll Your Own" Transformer that is!
With this project, you design and wind a transformer and use it to get +12V and -12V from a 9V battery from the DC-to-DC converter.

Subscriber's Price **$37.49**
Non-Subscriber's Price **$39.95**
Extra Cores, and Bifilar can be bought separately.

NixieNeon Clock Kit

Sale!



Using Neon and Nixie bulbs, this clock displays the precision movement of time and is programmable to display "Time Chaos" at different intervals. The NixieNeon Clock then resets itself to the correct time in an eye-catching ballet of luminosity. The PCB is 7.25"x 7.25" and consists of over 400 components.

Reg $195.95
Sale Price $159.95
PCBs can be bought separately.

Now With New Code!

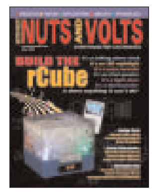
Transistor Clock Kit

If you like electronic puzzles, then this kit is for you! There are no integrated circuits; all functionality is achieved using discrete transistor-diode logic. The PCB is 10"x11" and harbors more than 1,250 components! For more info, see page 42, this issue.

Reg $225.95
Sale Price $199.95
PCBs can be bought separately.

rCube Talking Alarm Clock KIt

As seen on the May 2009 cover

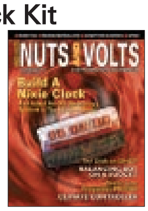


Available in blue, black, red, and green. All components are pre-cut & pre-bent for easier assembly and the microcontrollers are pre-programmed with the software. Kits also include PCB, AC adapter, and instructions on CD-ROM.

Subscriber's Price **$49.95**
Non-Subscriber's Price **$54.00**
PCBs can be bought separately.

Nixie Tube Clock Kit

Cherry Wood Clock Case



October 2006

Nixie tube clocks fuse the spirit, drama, and eerie beauty of cold war technology with modern inner works to create uncommon handcrafted timepieces.

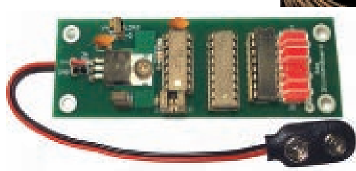
Now with optional case choices!
Get more info @ www.nutsvolts.com

Subscriber's Price **$146.95**
Non-Subscriber's Price **$159.95**
Kit includes article reprint, complete instructions and parts list.

Das Blinkenboard Kit

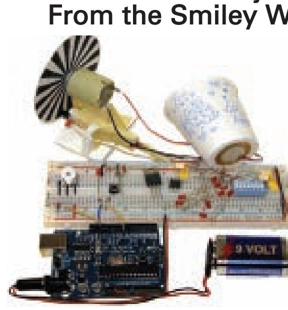
This kit includes a pre-programmed ATtiny84 microcontroller that sports eight software PWM channels to control motor speed and light brightness. Jumper selectable patterns can be used to operate motors, solenoid valves, relays.

Expand your board with the "Der Magnetfelder Detektor" componet pack.



Subscriber's Price **$32.45**
Non-Subscriber's Price **$35.95**

Arduino Project Kit From the Smiley Workshop



By Joe Pardue

Blink LEDs (Cylon Eyes). Read a button and 8-bit switch. Sense Voltage, Light, and Temperature. Make music on a piezo element. Sense edges and gray levels. Optically isolate voltages. Fade LED with PWM. Control motor speed and more.

$79.95

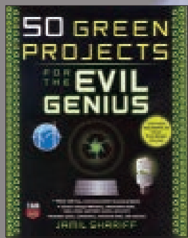
WE ACCEPT VISA, MC, AMEX, and DISCOVER. Prices do not include shipping and may be subject to change.

Attention **Subscribers ask about your discount on prices marked with an ***

The Nuts & Volts **WEBSTORE**

ALTERNATIVE ENERGY SECTION

50 Green Projects for the Evil Genius
by
Jamil Shariff



Using easy-to-find parts and tools, this do-it-yourself guide offers a wide variety of environmentally focused projects you can accomplish on your own. Topics covered include transportation, alternative fuels, solar, wind, and hydro power, home insulation, construction, and more. The projects in this unique guide range from easy to more complex and are designed to optimize your time and simplify your life!
$24.95

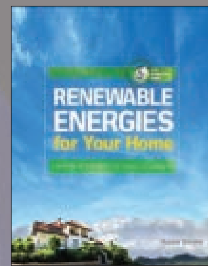
Do you know how many watts (YOUR MONEY) are going down the drain from "THE PHANTOM DRAW?"

If you are interested in your own power usage we at *Nuts & Volts Magazine* believe that this is the best way to help you determine your electrical energy use in ON and OFF home appliances.

To order call 1 800 783-4624 or online www.nutsvolts.com
$29.95 plus S&H

Renewable Energies for Your Home
by Russel Gehrke

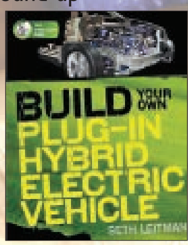
Simple and Inexpensive Renewable Energy Solutions for Your Home.



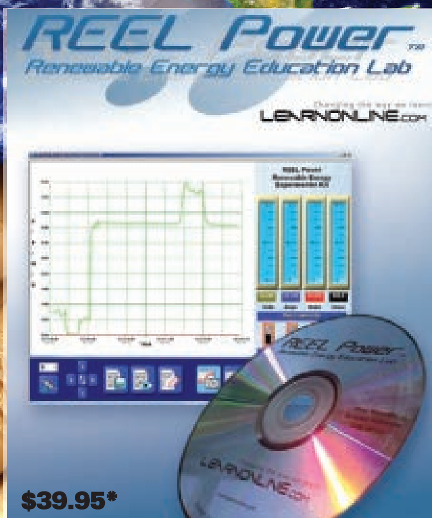
It's not always easy being green, and sometimes it costs more than you'd expect to get an eco-friendly home improvement project up and running. *Renewable Energies for Your Home* gives you sustainable home energy solutions that won't drain your wallet or the power grid. Filled with step-by-step instructions and helpful photos and illustrations. **$24.95**

Build Your Own Plug-In Hybrid Electric Vehicle
by Seth Leitman

A Step-by-Step Guide to Building a Plug-In Hybrid Electric Vehicle from the ground up



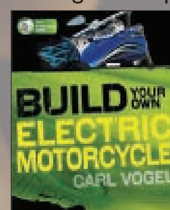
Build Your Own Plug-In Hybrid Electric Vehicle puts you in the driver's seat when it comes to hitting the road in a reliable, economical, and environmentally friendly ride. Inside, you'll find complete details on the hybrid powertrain and all the required components, including the motor, battery, and chassis.
$24.95



$39.95*

Build Your Own Electric Motorcycle
by Carl Vogel

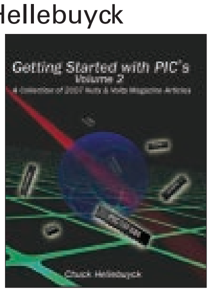
A practical guide for building an electric motorcycle from the ground up.



Complete with hundreds of step-by-step pictures, charts, and graphs for the latest and most efficient technologies, this new TAB Green Guru Guide shows you how to build an electric motorcycle from scratch. Written by an electric vehicle expert, *Build Your Own Electric Motorcycle* provides current data on all required materials, components, and specifications. **$24.95**

SPECIAL OFFER

Getting Started With PICs Volume 2
by Chuck Hellebuyck



Getting Started With PICs Vol 2 continues where the first book left off. This book contains the collection of the next 12 articles that were published in *Nuts & Volts* magazines of 2007. A great compliment to Vol 1 and a must have if you're entering the world of Pics & Progamming.
Reg $29.95
Introductory Price $24.95

ELECTRONET

For the ElectroNet online, go to

www.nutsvolts.com

click **Electro-Net**

■ BY FRED EADY

GO NUTS WITH THE KADTRONIX USB HID API LIBRARY

As a kid, I knew them as "hicker nuts." What my little Southern mouth was trying to say was hickory nuts. If you're a *Nuts & Volts* reader living in the southeastern United States, at one time or another you have probably been out there rooting around with the squirrels collecting hickory nuts.

For those of you that don't know what a hickory nut is, it is a cousin to the pecan and grows wild in my hometown of Fayetteville, TN. A typical Tennessee hicker nut is a bit smaller than a quarter and is contained within a very hard outer shell. As a kid, my mother and I would go into the woods on my grandma's farm in search of hicker nuts armed only with a couple of claw hammers. Hickory nuts have to be shelled with a hammer and a rock as the casings are too hard to crack with your hands or teeth. We would eat the good ones we found on the spot, then move on to the next prolific nut location once we had rooted out all of the good nuts. The tell-tale sign of a bad hicker nut is a small hole drilled into the shell by a worm or other strong-mouthed insect. Our foraging would last for hours since hickory nut meat is very small, and every hammer smack didn't guarantee that an edible nut was waiting inside.

Hunting hicker nuts is very similar to wading through Microsoft Visual C++ code. There's lots of hard outer shell and cracking through doesn't always mean you'll find what you're looking for. You also have to know where the trees are. If you can't find the trees, you can't pick up nuts. You can't crack the nuts if you can't find them. If you do manage to find some nuts, you'd better have the correct tools to crack them.

THE KADTRONIX USB HID API LIBRARY

The Kadtronix USB HID API Library was initially designed to support Visual Basic 6 and Visual C++ 6. Later, the Kadtronix folks wrote a version that supports the .NET version of Visual C++. Although you can still purchase Visual Basic 6 and Visual C++ 6 from vendors on the Internet, at the time of this writing the latest and greatest versions of Visual Basic and Visual C++ could only be had in Microsoft's Visual Studio 2008 product. (Visual Studio 2010 is supposedly due to be released in March.) So, rather than base this month's discussion on the legacy Visual Basic and Visual C++ compilers, I decided to show you how to move data between a PC and our PIC32MX795F512L TRAINER with the compilers contained within the Microsoft Visual Studio 2008 Standard. This is possible due to the Kadtronix Library's affinity for compilers that can access dynamic link libraries (DLLs).

After a couple of days attempting to use the Visual Studio version of Visual Basic with the Library, I found that it and the .NET-based Visual Basic component of Visual Studio 2008 are incompatible as they stand. Most of my grief was created by the differences in the ways Visual Basic 6 and the .NET Visual Basic component handle string data. The API Library uses Visual Basic 6 string handling methods that just don't exist in the .NET component of Visual Studio 2008. The Kadtronix documentation points out that the API Library may not support the .NET Framework in the Visual Basic environment. I felt that I owed it to the Design Cycle readers to put the Visual Studio version of Visual Basic to the test myself. The Kadtronix folks were right on the money.

The Kadtronix Library package includes a .NET DLL (UsbHidApi_NET.dll) for Visual C++. However, I was able to combine the Visual Studio 2008 version of Visual C++ and the standard Visual C++ Kadtronix Library DLL (UsbHidApi.dll) to produce the HID data transfer code that we will be discussing.

The UsbHidApi.lib file (which is included with the Library package) is also a necessary part of the Visual C++ compilation and link process. I explicitly defined the UsbHidApi.lib to the Visual C++ linker from within the VS 2008 configuration controls. The UsbHidApi.dll file is embedded within the Windows/system32 directory during

the installation of the Library package. Okay, now that you know where the trees are, let's crack some nuts.

TOPSEY TURVEY

Normally, we would discuss the microcontroller side of things before adding the PC ingredients into the mix. Instead, let's start by sorting through the Microsoft hicker nuts. The first nut we come across is the Kadtronix UsbHidApi.lib file which contains the interface functions and definitions that reach into the bowels of the Windows HID service engine. This file allows us to easily manipulate Windows' built-in HID features without an in-depth knowledge of Windows OS programming. As I mentioned earlier, the UsbHidApi.lib file is supported by a DLL which contains other important data items and functions that will shield us from the heat of raw Windows programming.

We've all heard the stories about desperate firmware thieves slithering off the packages of code-protected microcontrollers and stealing their code by examining the exposed silicon with a microscope. Once they had a "photograph" of the microcontroller's silicon layout, they could reverse-engineer the protected functions. A like situation exists with a .lib or .dll file. If we just absolutely had to know how the library or DLL works, we could use some expensive software tools to break it down and analyze the assembler code. But who really wants to do that? After all, obtaining the tools means nothing if you don't have the skills to interpret and rewrite the code. If the would-be firmware cheat had the knowledge to reverse-engineer the stolen assembler mnemonics, he or she wouldn't need to steal the code because they could write the necessary algorithms from scratch.

Unlike that code-protected microcontroller, we have a road map that helps us navigate the .lib and .dll functionality of the Kadtronix Library. That road map is in the guise of a file called UsbHidApi.h which is part of the Kadtronix install package.

The UsbHidApi.h file contains the information we need to be able to access the internal functionality of the UsbHidApi.lib and UsbHidApi.dll library files. For instance, here are the Read and Write HID library function declarations found within UsbHidApi.h:

```
extern "C" int _stdcall Read(void *pBuf);
    // Read from the HID device
extern "C" int _stdcall Write(void *pBuf);
    // Write to the HID device
```

The Library also houses other types of data declarations in the UsbHidApi.lib and UsbHidApi.dll files which include character and integer variable definitions, structures, and inheritable class instances.

Although we will only be addressing a single HID-class device, the API Library has the ability to query and identify multiple HID-class devices. Each device's parameters are stored in a structure called mdeviceList which is revealed to us in UsbHidApi.h. Here's what the mdeviceList structure looks like:

```
typedef struct {
  char       DeviceName[50];
     // Device name
  char       Manufacturer[50];
     // Manufacturer
  char       SerialNumber[20];
     // Serial number
  unsigned int VendorID;
     // Vendor ID
  unsigned int ProductID;
     // Product ID
  int        InputReportLen;
     // Length of HID input report (bytes)
  int        OutputReportLen;
     // Length of HID output report (bytes)
  int        Interface;
     // Interface
  int        Collection;
     // Collection
 } mdeviceList;
```

Instead of me running my mouth about the mdeviceList structure, let's write some C code to fill its variables with information we obtain from the PIC32MX795F512L TRAINER.

THE GETLIST API CALL

If multiple HID-class devices need to be uniquely serviced by our HID host, we must be able to identify each of them individually. That's where the GetList API call comes in:

```
int GetList(unsigned int VendorID,
              // Vendor ID to search
              // (0xffff if unused)
           unsigned int ProductID,
              // Product ID to search
              // (0xffff if unused)
           char          *Manufacturer,
              // Manufacturer (NULL if unused)
           char          *SerialNum,
              // Serial number to search
              // (NULL if unused)
           char          *DeviceName,
              // Device name to search
              // (NULL if unused)
           mdeviceList  *pList,
              // Caller's array for storing
              // matching device(s)
           int           nMaxDevices);
              // Size of the caller's array list
              // (no.entries)
```

The GetList function is part of the class imported from the UsbHidApi DLL called CUsbHidApi. If an attached HID-class device is available, invoking GetList will pull its operating parameters into a slot of the mdeviceList structure array. Before we can invoke GetList, we need to bring an instance of CUsbHidApi called hidDevices to life. The name hidDevices is arbitrary:

```
CUsbHidApi    hidDevices;
```

Using the mdeviceList structure outlined in the UsbHidApi.h file as a template, we must also create an array of structures which we will call m_DeviceList. Note that there are only two device structures available in our m_DeviceList[2] array. If we needed to track more than two devices, we would simply allocate the necessary

number of structure array entries. Right now, we are limited to detecting only two devices. In reality, we only require a single structure in our array. Our GetList call will reside inside of the DetectDevice function. So, we'll also code the DetectDevice function prototype into dcHidApidlg.h:

```
//declared public in dcHidApiDlg.h
mdeviceList m_DeviceList[2];
void DetectDevice(void);
```

The next step involves declaring the variables that the GetList function requires. The variable declaration process includes initialization of *pList which is loaded to point to the array of structures we created in dcHidApiDlg.h. Every variable declared can be traced back to its roots in the GetList function that was imported from the UsbHidApi.dll:

```
        //this code is part of dcHidApiDlg.cpp
        void CdcHidApiDlg::DetectDevice(void)
{
                unsigned int activeDevices,
                vendor_id,product_id;
                char  *manufacturer,*serial_num,
                *device_name;
                mdeviceList *pList = m_DeviceList;
```

Once we have declared all of the variables we need for the GetList function, we prepare the GetList function to get information on every HID-class device that is available by not specifying any information in the call that can be traced to any particular HID-class device that has made itself available:

```
vendor_id = 0xFFFF;
        // 0xFFFF => Any vendor ID
product_id = 0xFFFF;
        // 0xFFFF =>Any product ID
serial_num = NULL;
        // NULL=> Any serial number
device_name = NULL;
        // NULL=> Any device name
manufacturer = NULL,
        // NULL=> Any manufacturer
```

Upon invocation, the GetList API function will return the number of HID-class devices that respond to the function's request. In addition, each device that responds will fill a structure entry in the array of structures in m_DeviceList which is referenced by the pointer *pList:

```
activeDevices = hidDevices.GetList(
   vendor_id,              // Vendor ID
   product_id,             // Product ID
   manufacturer,           // Manufacturer
   serial_num,             // Serial number
   device_name,            // Device name
   pList,                  // Device list
   2);                     // max number of
                           // devices
```

The data that flows from a responding device to the m_DeviceList structure originates in the TRAINER's device descriptor.

DESCRIPTOR REVIEW

Before we execute the GetList API function, let's grease the skids and take a look at what to expect in the TRAINER's m_DeviceList structure entry. As I mentioned earlier, every piece of data returned to the m_DeviceList structure will be gleaned from a PIC TRAINER descriptor entry. The Microchip MCHPFSUSB Framework takes the hard work out of descriptor creation.

The first element of our m_DeviceList structure is the DeviceName which is a 50 character array. I entered the following device name into the TRAINER descriptor file:

```
//Product string descriptor
ROM struct{BYTE bLength;BYTE bDscType;WORD
string[12];}sd002={
sizeof(sd002),USB_DESCRIPTOR_STRING,
{'O','U','R','H','I','D','D','E','V','I','C','E'
}};
```

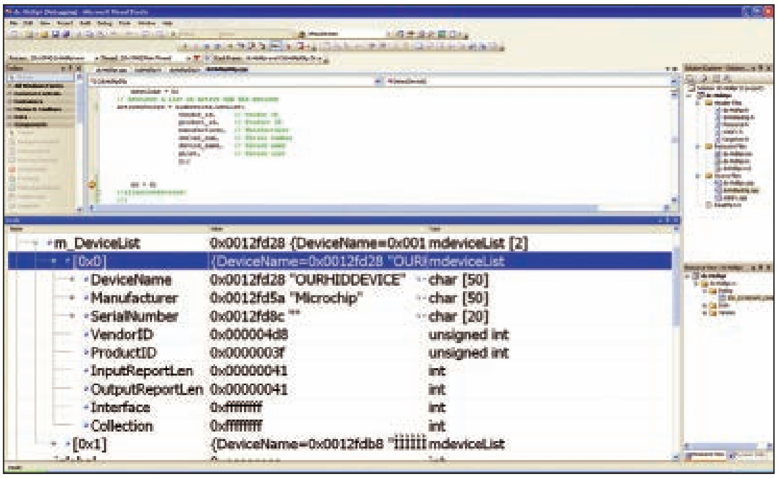
The VID entered in the descriptor can be used to trace the manufacturer of the device. However, we can also specify the manufacturer with a string in the device descriptor:

```
//Manufacturer string descriptor
ROM struct{BYTE bLength;BYTE bDscType;WORD
string[9];}sd001={
sizeof(sd001),USB_DESCRIPTOR_STRING,
{'M','i','c','r','o','c','h','i','p'
}};
```

There is no serial number entry in the descriptor. Therefore, we should expect an empty serial number structure entry to be returned from the TRAINER. If you require a serial number, the header text of the MCHPFSUSB Framework usb_descriptor.c file can be used as a guide to adding the necessary serial number descriptor code.

This whole process would be useless without a PID and VID entry. So, here are the Vendor ID (VID) and Product ID (PID) entries that I entered into the TRAINER's descriptor:

```
ROM USB_DEVICE_DESCRIPTOR device_dsc=
{
    0x12,       // Size of this descriptor
                // in bytes
    USB_DESCRIPTOR_DEVICE,
                // DEVICE descriptor type
    0x0200,     // USB Spec Release Number
                // in BCD format
    0x00,       // Class Code
    0x00,       // Subclass code
    0x00,       // Protocol code
    USB_EP0_BUFF_SIZE,
                // Max packet size for EP0,
                // see usb_config.h
    0x04D8,     // Vendor ID
    0x003F,     // Product ID
    0x0002,     // Device release number in
                // BCD format
    0x01,       // Manufacturer string index
    0x02,       // Product string index
    0x00,       // Device serial number string
                // index
    0x01        // Number of possible
                // configurations
};
```



■ SCREENSHOT 1. This is a capture of the data that flows from the PIC32MX795F512L TRAINER into the m_DeviceList structure entry. Note that the second entry is available to us but not put to use.

Since I haven't coughed up a fee for my own set of USB IDs, I've taken the liberty to use the Microchip VID coupled with a PID that is associated with the cursor-in-a-circle demo program.

If you examine the usb_descriptors.c file in the download package (available at **www.nutsvolts.com**), you'll find that only one interface and a default collection are coded. Thus, don't expect any unique collection and interface numbers to be returned. Within the collection descriptor area, you'll see that the report size is set as 64 bytes. That means 65 bytes maximum are sent along inside of a report. The 65th byte is the Report ID header byte which resides at the beginning of the report package.

Now that we know what to look for and what to expect, the results of our GetList API call can be viewed in **Screenshot 1**.

VISUAL C++ TO PIC32MX795F512L TRAINER

The TRAINER responded to the GetList API call and as a result, the variable activeDevices assumed a value of 0x0001. At this point, if multiple devices responded, we could choose the device that we wish to communicate with from our list stored in the m_DeviceList array. Since activeDevices is equal to 1, there is only one device we can talk to.

To communicate with the TRAINER, we must open the device. To do this, we use the GetList parameters returned from the TRAINER and the Kadtronix Library Open API function:

```
rc = 0;
if(activeDevices)
{
       rc = hidDevices.Open(
     pList->VendorID,
     pList->ProductID,
     pList->Manufacturer,
     pList->SerialNumber,
     pList->DeviceName,
     TRUE);    // Use non-blocking reads
}
```

We can use any of the Open API call arguments to identify the device we wish to open. Unused Open API function argument identifiers are loaded with NULL or 0xFFFF values. In our case, we're using every variable that was returned to us by the TRAINER as an identifier. The pointer *pList is currently pointing to the first entry of the m_DeviceList array. If we needed to look at the second structure in m_DeviceList, we would simply increment *pList.

I've put together a little application that will communicate its status and any data we request via a Visual C++ ListBox. Our GetList API call was successful and the variable activeDevices is used as the key to the Open API call. If the Open API call returns a TRUE (0x0001), the link to the PIC32 TRAINER was successfully opened. The code snippet that follows does all of the talking for the Open API call:

```
CString csVendorID, csProductID;
CString csInputReportLen, csOutputReportLen;
CString OPENOKtxt, OPENERRtxt;
unsigned int rc;

if (rc)
{
       OPENOKtxt.Format(_T("Device Opened"));
       m_DisplayWindow.InsertString(nextline++,
       OPENOKtxt);
       csVendorID.Format(_T("VID = 0x%0.4X"),
       pList->VendorID);
       m_DisplayWindow.InsertString(nextline++,
       csVendorID);
       csProductID.Format(_T("PID = 0x%0.4X"),
       pList->ProductID);
       m_DisplayWindow.InsertString(nextline++,
       csProductID);
       csInputReportLen.Format(_T("Input Report
       Length = %d"), pList->InputReportLen);
       m_DisplayWindow.InsertString(nextline++,
       csInputReportLen);
       csOutputReportLen.Format(_T("Output Report
       Length = %d"), pList->OutputReportLen);
       m_DisplayWindow.InsertString(nextline++,
       csOutputReportLen);
}
else
{
       OPENERRtxt.Format(_T("Device Open
       FAILED"));
       m_DisplayWindow.InsertString(nextline++,
       OPENERRtxt);
}

       return;
}
```

Screenshot 2 is a result of the positive response to the Open API function call. The Open API call did not change the contents of the m_DeviceList array. So, we can display the PIC32MX795F512L TRAINER GetList values with our positive link message.

I failed to mention that I had a little HID analog-to-digital (A-to-D) conversion routine in my pocket. So, I'm

sure you're wondering what clicking on the Read Voltages button does. Once again, I'll shut up and just show you the code:

```
void CdcHidApiDlg::OnGetVolts()
{
       char xmit_buf[100];
       char recv_buf[100];
       unsigned int wrc,rrc;
       float raw_volts;
       CString vReading;

       memset(xmit_buf, 0,
       sizeof(xmit_buf));
       xmit_buf[1] = 0x37;
       wrc = hidDevices.Write(xmit_buf);
```

The partial code snippet of the OnGetVolts function creates a pair of 100 byte buffers. The xmit_buf array is preloaded with zeros which automatically puts our desired Report ID (0x00) in the first report buffer slot. The Report ID is immediately followed by a command byte of 0x37 which corresponds to a C case statement in the download package file nv-pic32mx-HID.c:

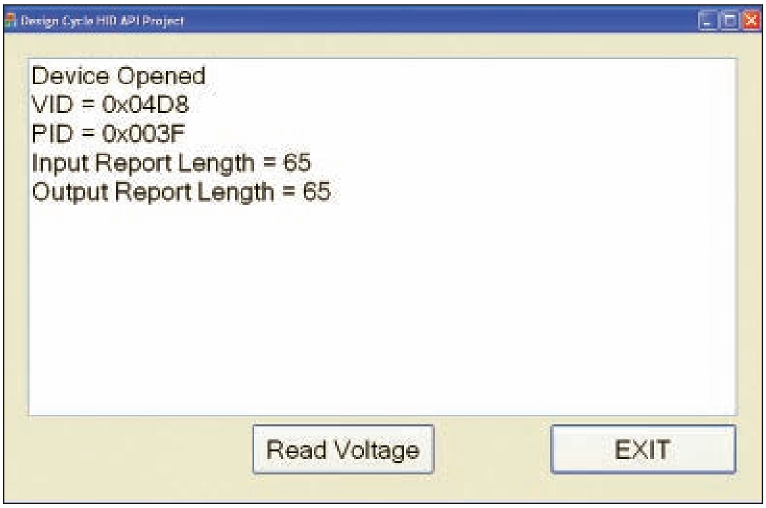
```
case 0x37:    //Read POT command
  {
    WORD_VAL w;

    if(!HIDTxHandleBusy(USBInHandle))
        {
           mInitPOT();
           w = ReadPOT();
             // Use ADC to read the
             // I/O pin voltage.
           ToSendDataBuffer[0] = 0x37;
             // Echo back to the host
           ToSendDataBuffer[1] = w.v[0];
             //Measured analog voltage LSB
           ToSendDataBuffer[2] = w.v[1];
             //Measured analog voltage MSB
           USBInHandle = HIDTxPacket
          (HID_EP,(BYTE*)&ToSendDataBuffer
          [0],64);
        }
     }
     break;
```

The ReadPOT function is configured to use RB2 as the analog input for the A-to-D converter. As you can see in **Schematic 1**, I've tied the wiper of a 10K potentiometer to the PIC32MX795F512L's RB2 pin. With this potentiometer configuration, the PIC32's A-to-D converter input will be presented with a minimum of zero volts and a maximum of 3.3 volts. Following the A-to-D conversion, the command byte and the raw voltage word are returned to the host's Visual C++ application. The modified hardware to support **Schematic 1** is lying under the lens in **Photo 1**.

PIC32MX795F512L TRAINER TO VISUAL C++

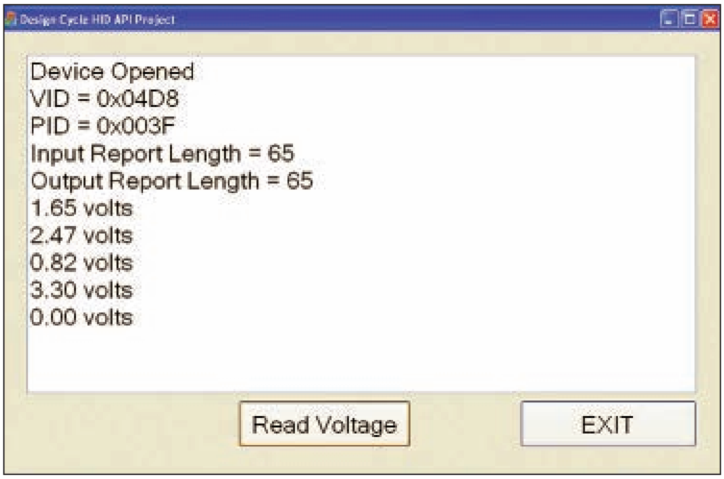
If all has gone as planned, the TRAINER received the 0x37 command and executed the A-to-D converter read operation against the potentiometer which is configured as a simple voltage divider. Approximately 100 mS later, the host issues a Read API function call to the open device which just happens to be our TRAINER. Here's the receive portion of the OnGetVolts function:

```
if (wrc == hidDevices.m_WriteSize)
{
  Sleep(100);
  rrc = hidDevices.Read(recv_buf);
  if(rrc == hidDevices.m_ReadSize)
       {
       raw_volts = (float)((recv_buf[2] >> 8)&
       0x00FF) + ((recv_buf[3] << 8) & 0xFF00);
       vReading.Format(_T("%2.2f
       volts"),raw_volts * .003222);
       m_DisplayWindow.InsertString
       (nextline++, vReading);
       }
}
}
```

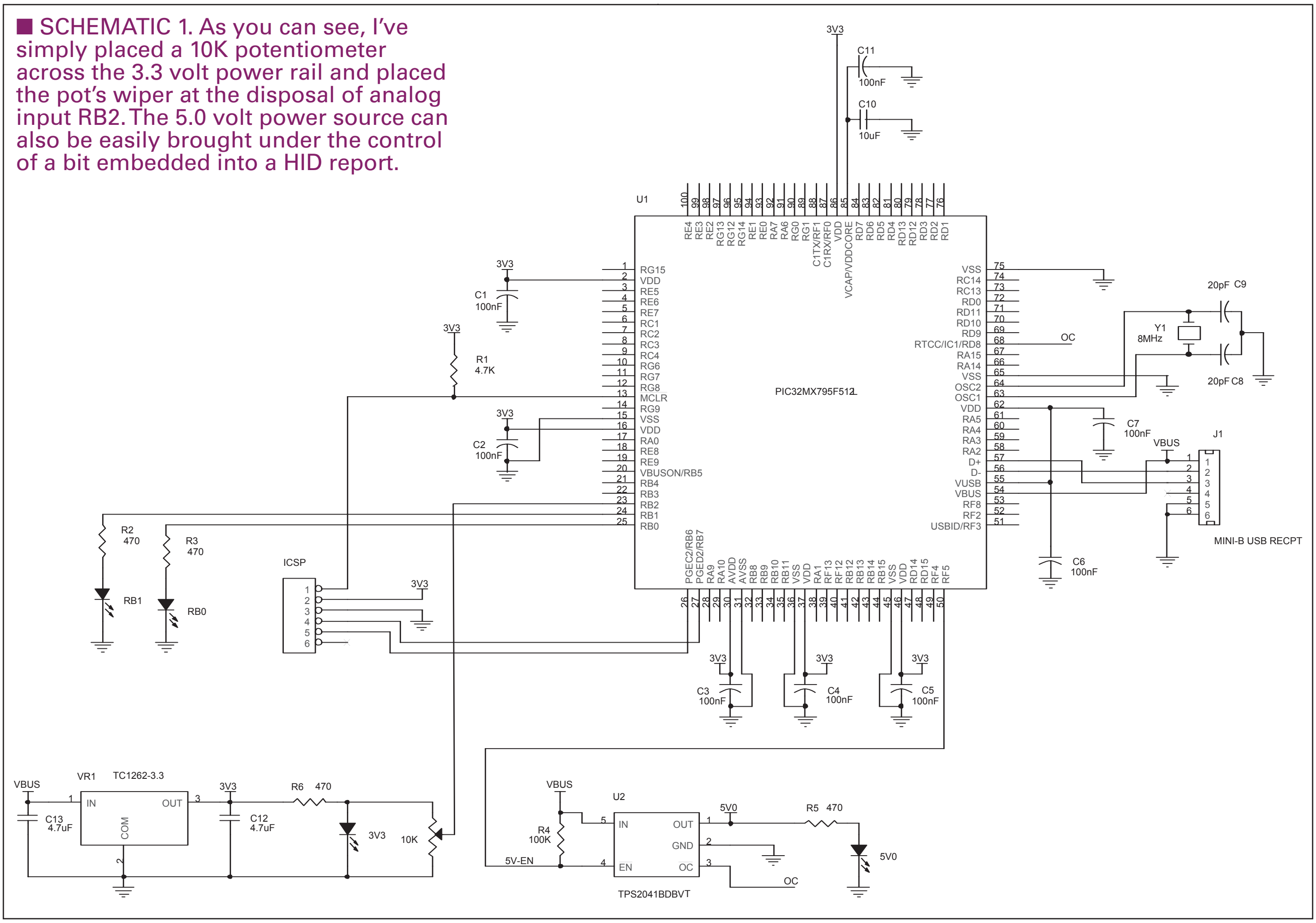


■ SCREENSHOT 2. Nothing fancy here. This is a simple Visual C++ ListBox control and a couple of buttons. The heavy lifting is being done by the Kadtronix USB HID API Library and some user-contributed Visual C++ event handlers.

■ SCREENSHOT 3. Each click returns a voltage reading. Everything in the ListBox with the exception of the Device Opened message is the direct result of HID report data flowing between the host Visual C++ application and the PIC32MX795F512L TRAINER.

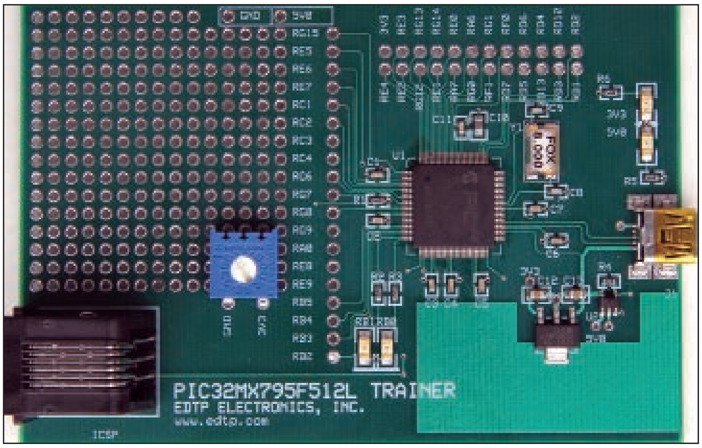


■ SCHEMATIC 1. As you can see, I've simply placed a 10K potentiometer across the 3.3 volt power rail and placed the pot's wiper at the disposal of analog input RB2. The 5.0 volt power source can also be easily brought under the control of a bit embedded into a HID report.



The Report ID byte and the echoed command byte precede the actual raw voltage data in the receive buffer. Since the 10 bits of incoming raw voltage data is in byte format, we need to convert the pair of voltage bytes to an integer. In that we must do some scaling of the integer voltage value for display in the ListBox, the integer voltage value must be converted to a floating point value. Twisting the potentiometer's wiper and clicking on the Read Voltage button produced the content contained within **Screenshot 3**.

■ PHOTO 1. This is a shot of my slightly modified PIC32MX795F512L TRAINER. The voltage from the potentiometer's wiper is one of many data types that can be transferred in a HID report. If you send a 0x80 instead of the 0x37, you'll gain control of the LEDs attached to RB0 and RB1.



AS EASY AS RS-232

I never thought I would admit it. With the MCHPFSUSB Framework working on the TRAINER side and the Kadtronix Library operating within Microsoft's Visual C++ component of VS 2008, coding a useful HID

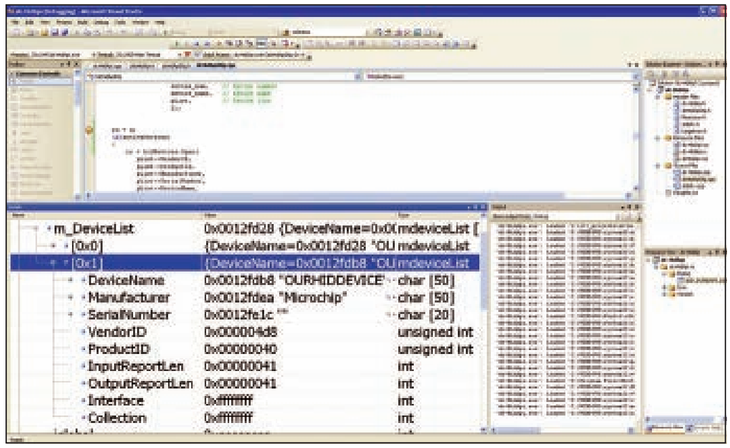
SOURCES

Microchip
MCHPFSUSB Framework ; PIC32MX795F512L
www.microchip.com

Microsoft
Visual Studio 2008
www.microsoft.com

Kadtronix
Kadtronix USB HID API Library
www.kadtronix.com

EDTP Electronics, Inc.
PIC32MX795F512L TRAINER
www.edtp.com



■ SCREENSHOT 4. We can easily add and communicate with more than one PIC32MX795F512L TRAINER by simply increasing the number of m_DeviceList entries to accommodate the number of uniquely identified TRAINERs.

data transfer application is just as easy as coding a similar RS-232 based program.

I couldn't leave you without putting a second PIC32MX795F512L TRAINER out there and trying to contact it. So, I changed the PID to 0x0040 and programmed a second TRAINER. You can see its second m_DeviceList entry in **Screenshot 4**. Boy, does that conjure up possibilities. I can see all of those light bulbs illuminating over your heads.

I'll put all of the good hicker nuts I found into the download package. Once you ramble through the basket of nuts, you can add host and HID-class device programming to your Design Cycle. **NV**

■ *Fred Eady can be contacted via email at fred@edtp.com and the EDTP Electronics website at **www.edtp.com**.*

GIZMOS THAT BRING YOUR DREAMS TO LIFE!

Laser and IR Sensors for your next security science project. 400nm to 1100nm

IR pin diodes .95 each

Phototransistors .95 each

20 and 28 pin ZIF sockets now in stock starting at $4.50

NIMH Battery Packs 8.4V 2000mAH $2.50

Lego Flashlights
Keychain $7.95
Head Lamp $11.95
7inch Lamp $19.95
Dynamo Lamp $21.95
Let that light shine in the next generation!

1000's of LED's in stock
1.8mm, 3mm, 5mm, and 10mm from $.40
1/2 watt $.95
1 watt Stars $3.60
RGB $1.50
UV, IR, Surface Mount

RGB LED Strips
12v LED Bars
☺
LCD Displays
Graphic and Character

Atmel ATmega8/168/328 Development Board
ZIF socket, switches, LED's, ISP, RS232, and more…
Only $35

Atmel STK500 compatible USB ISP programmer, 6 pin and 10 pin, works with AVR Studio
Only $38

Lighted switch $3.25
Micro-reed switches only $3.00
Limit switches starting at $2.00
Reset switches $.80
Aircraft toggle covers $1.80
Arcade game buttons in many colors $2.25

5797 NW Cornelius Pass Road
Hillsboro Oregon 97124

To see the best deals join our online mailing list or stop by our retail store!

www.SurplusGizmos.com
503-439-1249

Order online at:
www.melabs.com

Development Tools for PIC® MCUs
microEngineering Labs, Inc.

Phone: (719) 520-5323
Fax: (719) 520-1867
Box 60039
Colorado Springs, CO 80960

USB Programmer for PIC® MCUs
$89.95 (as shown)
RoHS Compliant
Programs PIC MCUs including low-voltage (3.3V) devices
Includes Software for Windows 98, Me, NT, XP, and Vista.
With Accessories for $119.95: Includes Programmer, Software, USB Cable, and Programming Adapter for 8 to 40-pin DIP.

LAB-X Experimenter Boards
Pre-Assembled Board
Available for 8, 14, 18, 28, and 40-pin PIC® MCUs
2-line, 20-char LCD Module
9-pin Serial Port
Sample Programs
Full Schematic Diagram
Pricing from $79.95 to $349.95

BASIC Compilers for PICmicro®
Easy-To-Use BASIC Commands
Windows 98/Me/2K/XP/Vista
PICBASIC™ Compiler $99.95
BASIC Stamp 1 Compatible
Supports most 14-bit Core PICs
Built-In Serial Comm Commands
PICBASIC PRO™ Compiler $249.95
Supports most PICmicro® MCU Families
Direct Access to Internal Registers
Supports In-Line Assembly Language
Interrupts in PICBASIC and Assembly
Built-In USB, I2C, RS-232 and More
Source Level Debugging

EPIC™ Parallel Port Programmer
starting at $59.95

PICPROTO™ Prototyping Boards
Double-Sided with Plate-Thru Holes
Circuitry for Power Supply and Clock
Large Prototype Area
Boards Available for Most PIC® MCUs
Documentation and Schematic
Pricing from $8.95 to $19.95

See our full range of products, including books, accessories, and components at:
www.melabs.com

CLASSIFIEDS

KITS/PLANS

High Quality
Electronic Kits & Modules
$29.95
KIT
CK1122
Dual Bidirectional
H-Bridge Motor Controller
FM Transmitters
Timers
Audio Amplifiers
Motor Controllers
PIC Programmers
••• Custom Designs •••
www.canakit.com

50 MHz auto ranging frequency Counter. Pic microprocessor.
Viewing from 1Hz-50MHz
Complete kit
$49.95 + $5.95 S&H
Order: PayPal
www.Invent-Electronics.com
Or: 304-838-6293

NEW FOR 2010
VIDEO AGC/SYNC PROCESSING KIT
All Board Mounted Parts
w/ Instruction Manual CD
Choice of Non-programmed or Pre-programmed AVR Microcontroller
www.jlkelectronics.com

Electronic Kits
• Fun and Educational
• Discount White and Blue LEDs
• Quantity Discounts
www.bakatronics.com

QKITS.COM
Qkits Ltd
50 AMP PWM
1-888-GO 4 KITS

HARDWARE WANTED

DEC EQUIPMENT WANTED!!!
Digital Equipment Corp. and compatibles.
Buy - Sell - Trade
CALL KEYWAYS 937-847-2300
or email buyer@keyways.com

MISCELLANEOUS FOR SALE

MUST SELL ...
1000 small ac/dc motors $1.00; 10,000 CTS tweeters $0.20; new car speakers with grilles made in USA $1.80; large inventory of speakers and speaker parts; $100,000 of all kinds of light bulbs.
Everything at a fraction of cost.
PH: 956-444-0004 (toll free) 888-351-5550
list and pictures on our web: www.iseliquidator.com

LCDs/DISPLAYS

4 1 1 Technology Systems
20 x 4
As low as $10.00ea w/backlight
16 x 2
16 x 2
More LCD's at
We even have TFT LCD's!
www.411techsystems.com

GADGETS

GREEN LASER POINTER
.93 mile range
$22.89 Free Shipping
www.gadgetcentral.ca

OSCILLOSCOPES

7 in 1 Scope !
CircuitGear CGR-101™ is a unique new, low-cost PC-based instrument which provides the features of seven devices in one USB-powered compact box: 2-ch 10-bit 20MS/sec 2MHz oscilloscope, 2-ch spectrum-analyzer, 3MHz 8-bit arbitrary-waveform/standard-function generator with 8 digital I/O lines. What's more – its open-source software runs with Windows, Linux and Mac OS's! **Only $180**
Saelig.com unique electronics
1-888-7SAELIG
info@saelig.com
www.saelig.com

SURPLUS

SURPLUS ELECTRONIC PARTS & ACCESSORIES
Over 13,000 Items in Stock.
Cables, Connectors, Displays, Fans
Hardware, LEDs, Motors, Potentiometers
Relays, Semiconductors, Service Manuals, Speakers
Switches, Test Equipment, Tools, VCR Parts
Surplus Material Components
SMC ELECTRONICS
www.smcelectronics.com
No Minimum Order.
Credit Cards and PAYPAL Accepted.
Flat $4.95 per order USA Shipping.

Surplus - Obsolete - Used
electrical or electronic
find it here or sell it here !
no fees ~ no commissions ~ it's free !
www.SecondHandRadio.com

BOOKS

Books
PIC® Microcontroller
sq-1.com
Stepper Motors
stepperstuff.com
CNC
cncintro.com
SQUARE 1 ELECTRONICS
(208) 664-4115
Hayden, ID USA

SOLAR PRODUCTS/KITS

BATTERY DESULFATORS
www.wizbangplus.com

PCB SOFTWARE

EXCLUSIVE U.S. DISTRIBUTOR
Circuit Wizard
YOUR COMPLETE SOLUTION!
• PCB Design
• PCB Simulation
• CAD/CAM Menu
121747, Hobby Vers., $175
121743, Pro. Version, $275
www.KELVIN.com

DESIGN/ENG SERVICES

* Circuit board layouts
* Prototype assemblies
WWW.OSPREYELECTRONICS.COM
Convert your sketch or print into a quality pcb for a reasonable price. Visit us on the web or call Osprey Electronics at (208) 664 1089 (USA)

FLEXREV China Direct Manufacturing
Turn Key or Subassemblies
Components, Enclosures, Keypads
$0.007 per soldering point
2 layer PCBs $0.14/in2
Small jobs welcome
FLEXREV.COM

PROFESSIONAL SERVICES

RICHARD N. KRAJEWSKI
Editor
201-659-0041
richardkrajewski.com

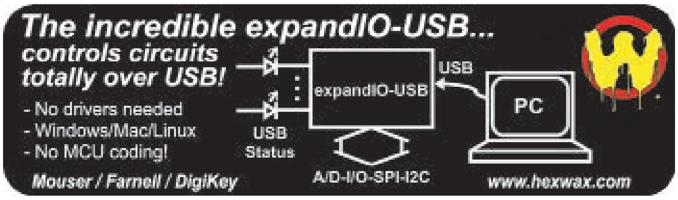
ROBOTICS

$29.95 (MSRP) MaxSonar-EZ1
High Performance Ultrasonic Range Finder
- serial, analog voltage & pulse width outputs
- lowest power - 2mA
- narrow beam
- very easy to use!
www.maxbotix.com

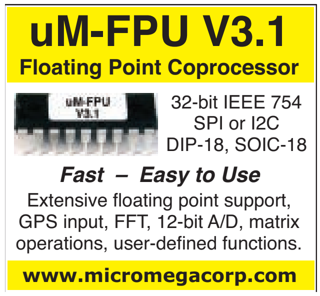
Looking for robots?
www.servomagazine.com

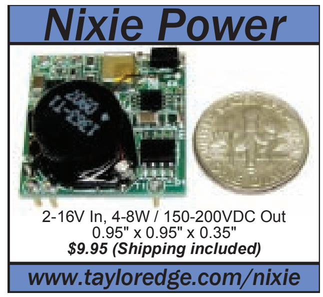
CLASSIFIEDS

MICROCONTROLLERS



COMPONENTS





FEEDBACK

continued from page 9

SPIN may not be as useful as learning C if you plan to work with a wide range of microcontrollers from different vendors. However, with the recent releases of C compilers for the Propeller, this relieves you of the requirement of learning SPIN just to use it. On the flip side, there are some who have adopted SPIN and have become both staunch supporters and advocates of the language.These folks have tirelessly worked to make "objects" available on the Parallax object exchange that take the heavy lifting out of writing SPIN code for the Propeller. These objects hide the complexity of using common devices with the Propeller such as keyboards, mice, NTSC monitors, and VGA video displays. In case you haven't had a chance to browse the library of available objects, have a look at ***http://obex.parallax.com/****. So, with that being said, let's look at other criteria that might influence your decision. In your case, it seems you have an excellent ability to absorb new ideas and information. Learning about SPIN might actually be a fairly simple task for you. You mentioned an interest in assembly language, and it should be noted that many of the more time-sensitive and/or difficult tasks that have been done on the Propeller have used assembly language to make them a reality. Bottom line is that (IMHO) I think you should get yourself a Propeller demo board (or maybe even a Stingray robot) and have a go at it. It seems well within your means and it certainly is a very fun and interesting new chip offering that iswell supported by its adopters and the manufacturer. I hope this answers your questions. Thanks again for writing. Please let me know how things progress for you.*

Vern Graner

OUT TO LAUNCH(ER)

Great column in *Nuts & Volts* on the Stingray! I enjoyed it. Figure 13 (which showed a toy launcher mounted on the Stingray) peaked my interest. I am not aware of a method for interfacing with the launcher without a PC. Can you let me know if you know how it can be done or if you know where I can find the information? I'd be interested in trying this with an Arduino or the Stamp. Thanks.

Bukasa

Hi Bukasa! I'm glad you enjoyed the article on the Stingray robot. In answer to your question about interfacing the USB missile launcher to the Propeller, I actually haven't done that yet. In the figure you see at the end of the article, I had simply placed the missile launcher on top of the robot for the photo. I haven't had the time to electronically connect it yet. As "speaking" USB is comparably difficult (and really unnecessary for the focus of this project), I've decided to gut the missile launcher and drive it directly from the Propeller outputs.

I haven't opened the missile launcher to see exactly how it is wired inside but — based on observations of the unit when operated by the USB cable — it appears there are simply three motors in the unit: Motor 1: Pan (left<-->right); Motor 2: Tilt (up<-->down); and Motor 3: Fire Missiles.

The Motor #3 goes through a cycle releasing a single missile, and then "parks" afterward. As such, there may be a switch or sensor that would have to be watched to determine when to stop the firing motor. All of the motors in the missile launcher are powered by three 1.5V AA batteries in the base of the unit, so we know the motors will work well with 4.5 volts. In order to control these from a Stingray, I'll probably need to:

1) Provide a power source. This could be done by leaving the original batteries in the missile launcher unit or by using the 5V switching power supply on the Stingray's MSR1 board as it is rated to produce 5V at up to three amps. I plan to check to make sure the motors don't produce noise that may interfere with the MSR1 though! 2) Provide motor control. I plan to disconnect the motors from whatever internal driver board is inside the missile launcher, and then connect the motors directly to some H-bridges like the SN754410 quad half H-bridge, available from SparkFun Electronics at ***www.sparkfun.com/commerce/product_info.php?products_id=315*** *or use the L293D available from Parallax at* ***http//tinyurl.com/yf8u9ys****. Of course, a simpler way (but more expensive) would be to use some serially operated motor controllers like these from Pololu at* ***www.pololu.com/catalog/product/410****. These units can control two motors each, so the project would need at least two of them, so I may not go that route. 3) Write the software! Lastly, after I manage to get the motors interfaced to the Propeller, I'll need to write the software to control them ... but that's a completely different topic! I hope the above information is useful. Please let me know if you decide to get a Stingray and/or if you manage to build a missile launcher into it!*

Vern Graner

TECHFORUM
READER-TO-READER

>>> QUESTIONS

Motion Sensing Faucet

I want to create a motion sensing electronic flow faucet. The solenoid valve is normally closed with an operating voltage of 24V.

I'm new to this hobby, so I'm not sure where to start as far as the circuit design goes. I thought of purchasing an electronic light switch and modifying it to trip the solenoid valve, but the switches are expensive.

#2101 **TJ Treinen**
Oklahoma City, OK

Spot Welder

Where can I find or make a simple spot welder suitable for rebuilding battery packs?

#2102 **John**

Centrally Monitor Door Status

I want to monitor the open or close status of 20 or more doors in a building. Are there devices made that would send a location address and magnetic door switch status via Cat 5 cable about 200′ back to a central panel? I would like these devices to share a common pair and be powered from the central panel too. The panel would light up an LED.

#2103 **Mike Carland**
Valencia, CA

AC Motor Control

I would like a diagram for a three-phase AC motor control that I can build myself. I can find all kinds for DC, but not AC. Could someone point me in the right direction?

#2104 **Sebastian Jaime**
WB4WLZR
Miami, FL

Welder Conversion

I am working on a project to enhance my Lincoln 225 amp 220 volt AC welder. I wish to find a circuit that can change the 60 Hz AC to a variable higher frequency. Welding aluminum with a tungsten torch (TIG) requires a variable frequency higher than the 60 Hz to make the best weld due to the nature of aluminum oxide and alloy. It is suggested in my reading that commercial TIG welder circuits change the sine wave to a square wave.

My welder has lugs on the transformer to switch to in order to select the range of amperage for welding steel. I will continue to set a range for the metal thickness of aluminum.

My concept is to add an auxiliary box to house the power supply for the electronics, the cooling fan, and either the cable to a foot switch or rheostat mounted to the torch head or surface mounting. A selector switch will turn the TIG feature on or off to continue to use the original welding feature or to TIG weld.

I have found circuits where the frequency can be varied, but at low voltage and amperage — nothing in the ranges I need. The components would be smoked immediately when switched on, or an arc was struck. One article on converting an AC to DC MIG welder suggested the diodes in the rectifier have a peak invert voltage of 1,000 volts for the momentary time when striking the arc, and be able to handle the amperage of the original welder — 225 amps. Although I am not going to convert my welder to DC, the information of peak voltage and amperage for any suggested circuit design may be helpful.

#2105 **John Johnson**
St. Clair Shores, MI

Battery Load Tester

I am looking for a circuit for testing the load of rechargeable batteries used in radios, cordless tools, etc. It must be switchable to allow testing of various voltages between 2V and 36V.

#2106 **Travis Branch**
Winston Salem, NC

Battery Substitution

I have a Victoreen 592b Geiger type device. It requires nine batteries; six of them are the 22.5V photo style battery, the other three are unknown. There is no part number or voltage listed, but the location for them makes them as 1-1/8" diameter and a little over 1/2" high.

I have searched the web and the Victoreen site but cannot find anything about any kind of battery fitting this description or any information about this device. Can anyone help with the information so I can use my old friend?

#2107 **Norm Doty**
Hollywood, FL

Vehicle Speed Threshold Trigger

To assist with parking, I'm installing front and side view cameras on my truck. The vehicle is already equipped with a rear-view camera (with display and associated electronics), and I also have a video switcher to select the desired view. As viewing any video while driving may be dangerous though, my goal is to disable the camera outputs at, say, 5 mph.

I'd like to use a PIC to monitor vehicle speed and output a signal when the speed threshold is met. The truck's PCM currently provides a 5V squarewave pulse signal which appears to be close to the actual vehicle speed (mph), e.g., 12 Hz = 10 mph. Any help utilizing a PIC to

All questions *AND* answers are submitted by *Nuts & Volts* readers and are intended to promote the exchange of ideas and provide assistance for solving technical problems. Questions are subject to editing and will be published on a space available basis if deemed suitable by the publisher. Answers are submitted by readers and **NO GUARANTEES WHATSOEVER** are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals.

Always use common sense and good judgement!

Send all questions and answers by email to **forum@nutsvolts.com**
*Check at **www.nutsvolts.com** for tips and info on submitting to the forum.*

achieve this goal would be greatly appreciated.

#2108 **MJR**
Cherry Hill, NJ

>>> ANSWERS

[#9094 - September 2009]
LED Ramp Up

I need a very basic schematic that will slowly ramp up an LED from minimum brightness to full and back. This is for the transmission towers on my model train table. By the way, thanks for the under track sensor schematic published in a previous month.

#1 For a very basic schematic, **Figure 1** is about as simple as it gets for analog LED dimmers. Using the ubiquitous TC555 timer operating in the astable mode provides the "ramp up" and "ramp down" times required by varying components C1 or R1. The varying voltage across C1 is buffered by the FET to drive the LEDs. The ZVN4306A was chosen mainly because that's what I had on hand and it worked well using red LEDs and only a three volt battery supply. You can substitute other FETs but may have to adjust the power supply voltage and/or the LED load resistor for best performance.

Steve Ghioto
Neptune Beach, FL

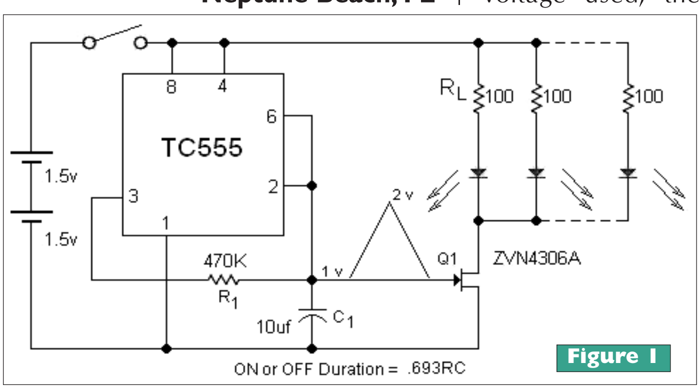


Figure 1

#2 A Microchip PIC10F202 (**Figure 2**) does a great job of simulating a tower light. By observing an actual tower, I determined that the following values work well:

Off Time
200 milliseconds
Fade Up Time
600 milliseconds
Bright Time
700 milliseconds
Fade Down Time
500 milliseconds

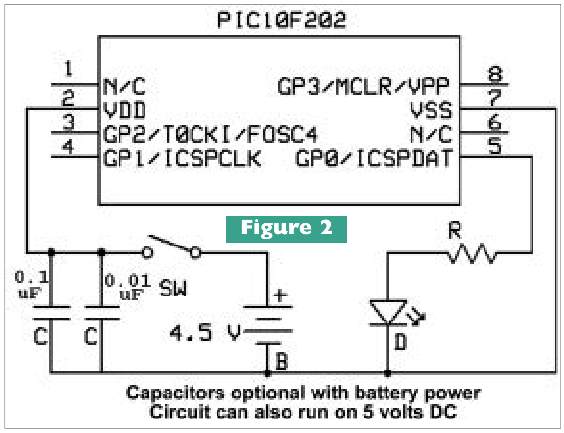


Figure 2

The document *U.S. Department of Transportation Federal Aviation Administration — Obstruction Marking and Lighting — ADVISORY CIRCULAR — AC 70/7460-1K* (Tower lighting) **https://oeaaa.faa.gov/oeaaa/external/content/AC70_7460_1K.pdf** gives details on tower lighting such as where (vertically) lights must be placed and which ones blink vs. which ones are continuously on and marking (painting) (how many color bands and what color for each). This document covers not only towers but ALL high obstructions such as buildings and water towers. The source code and hex code on the *NV* website implement these values.

The PIC10F202 will run with a supply of 2.0 to five volts, but six volts is above the Standard Operating Conditions. The resistor will need to be sized based on the supply voltage used, the voltage drop of the LED (different colors drop quite different voltages), and the desired current through the LED. A red LED will drop about 1.8 volts. The absolute max ratings for a port pin source or sink current is 25 milliamps. I could not find a recommended maximum, but I think 15 milliamps would be okay. Many red LEDs are quite bright with 10 milliamps, so for a five volt supply (allowing for a one volt drop at the output pin) I would suggest a 390 ohm resistor.

Mark Peterson
Plymouth, MN

#3 If you want to keep the parts count to a minimum, I suggest using a programmable device. Lately, I've been using PICAXE controllers in various projects. The PICAXE is a PIC with a built-in Basic interpreter.

The eight-pin PICAXE-08M is reasonably priced at around \$2.65. This device has several input and output ports. Some of the ports can be programmed for different uses.

The minimum circuit required to control an LED is shown in **Figure 3**. The LED is connected to a port configured for PWM (pulse width modulation) output. This means that

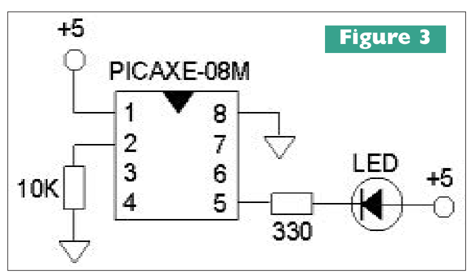


Figure 3

the port will output a waveform with a specific duty cycle. Changing the duty cycle will change the brightness of the LED. I was able to use the pwmout command to slowly ramp the brightness of an LED up and down with a short program. I could easily tweak the code to change the overall cycle time or rate at which the LED changes brightness.

Programming the PICAXE is easy using free software from **www.picaxe.co.uk**. There are several parts distributors in the US including **www.phanderson.com** who also sells a inexpensive prototype PC board kit with programming interface and voltage regulator for $8. A single PICAXE can be reprogrammed thousands of times, so it's easy to experiment and debug your program until you get it working the way you want.

The PICAXE can source or sink up to 25 mA per pin, so several LEDs could be supported by a single PICAXE circuit by wiring them in series and/or parallel. If you need to control more than four LEDs, it should be possible to sample the PWM output pin and drive the remaining output pins.

Code example:

```
  b0 = 0
  b1 = 0
myled:
  pwmout  2,50,b0
  pause  5
  if b1=0 then
   inc b0
   if b0=255 then
    b1=1
   endif
  else
   dec b0
   if b0=0 then
    b1=0
   endif
  endif
  goto  myled
```

Bob Kovacs
Barnegat, NJ

[#10091 - October 2009]
Radio Read Interrogation Device

My city has installed an Invensys water meter with a Sensus radio read transceiver. I would like to build/buy an interrogator device so I can transfer daily/hourly water use to my computer in order to monitor water consumption. Is this possible without buying an industrial reader?

If the Sensus water meter radio-read system is the same one we evaluated a few years ago for the city of Detroit, your only option is to install your own separate meter in series. There are models with pulses that represent units of flow, and some can generate an ASCII data stream of the dials on demand.

Most water systems have policies that do not allow customers to interface with their meters. This is their cash register, and their policy is "DO NOT TOUCH." They worry about setting precedents, so your chances of getting an approval for an interface are slim to none. Reading it without permission would be considered tampering because accessing the radio will deplete the battery that powers the radio.

The Sensus radio is powered by a battery expected to last many years. The radio idles in a low power monitor mode and is awakened by a signal from the passing drive-by reader vehicle, after which it transmits three reads at random intervals. The random transmissions are necessary because the wake-up signal will trigger all meters within range. It has been too long since this pilot project was scrapped and replaced by a fixed-base system, but I think the Sensus system (we tested several) had an acknowledge to prevent a re-trigger to save the battery and to allow missed meters to get through. The data format is proprietary, and it contains the meter's ID to distinguish it from your neighbor's. The reads are sorted by the reading system software. In summary, the radio cannot support additional reads. Daily reads would divide the battery life proportionately which would be unacceptable to the utility, and it is nearly impossible to restrict the signal enough to avoid triggering other near-by meters.

There is an interface that is read by a touch wand. The wand supplies

the power by induction, and the touch pad can be paralleled with the radio. I don't know if it can be retro-fit to your model Invensys meter. Again, the politics probably will block any chance of getting your utility to provide the device, and Sensus may not be willing to give you the information you need to simulate the touch reader. You probably could install your own meter for what the utility would charge to provide the touch pad.

Dennis Green, PE
Head Water Systems Engineer
Detroit Water & Sewerage Dept

[#11091 - November 2009]
Power Supply

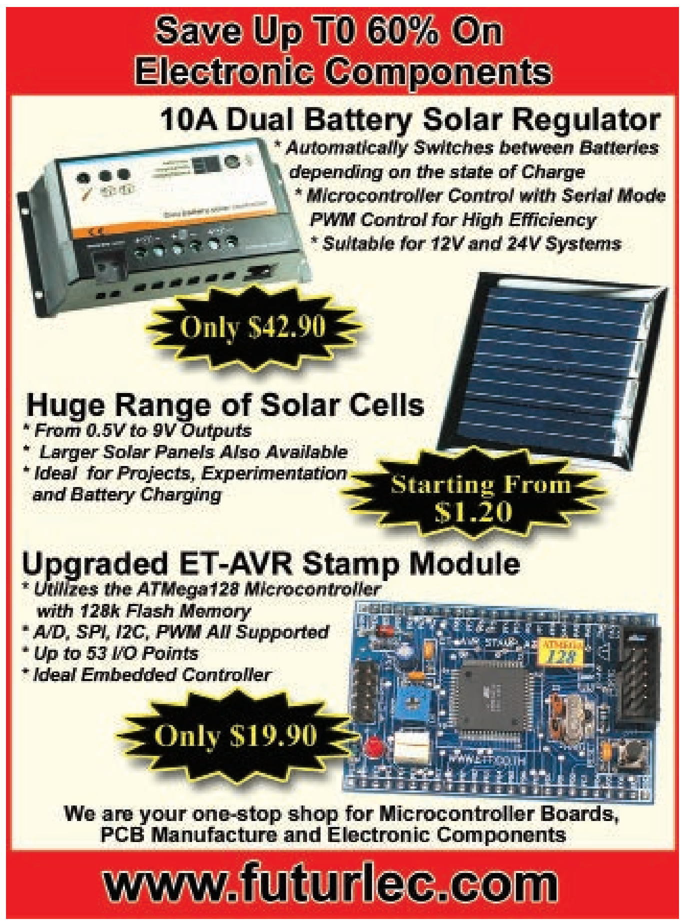
I am designing a project where weight is at a premium. I need some help designing a transformerless power supply that can deliver 12-14 volts at three amps. It needs to be auto switching 110/220 VAC. I designed a few projects in the past but they always had a transformer. I see things like laptop power supplies that don't have transformers. They put out three or four amps. How do they do it?

Your solution would be to use a switchmode power supply. This is the technology that allows small compact designs for low voltage, high current output power supplies.

Basic operation is as follows. Input of 90-240 VAC is rectified and smoothed somewhat by capacitors; the resulting DC is applied to a high-speed power MOSFET switching device whose gate is operated by a PWM controller at a high frequency (30-400 kHz). The power MOSFET drives the primary coil of a ferrite core transformer (yes, transformers are involved) at the frequency produced by the PWM controller; the secondary of the transformer is then connected to a rectifier diode/s to produce DC, and then capacitors and filters to smooth the DC and reduce high frequency components produced by the switching action. Feedback loops enable the PWM controller to regulate the output voltage by changing the duty cycle and/or frequency of the drive signal to the power MOSFET gate. The key to small compact size lies in the high switching frequency which enables the use of small ferrite core transformers.

Now to the designing of switchmode power supplies. This is not for the novice nor for someone inexperienced in the field of power electronics simply because of the high voltages involved and the critical nature of switchmode power supply design. If you want to find out more about switchmode power supplies, some good books and websites (**www.smpstech.com; www.smps.us**) are available. However, if you just need to get your project going, any electronics parts supplier (even surplus electronics suppliers) will carry ready-made and approved units in virtually any voltage or current combination to suit your needs.

Alberto Bonini
ONT, Canada



FRONT PANELS & ENCLOSURES
Customized front panels can be easily designed with our free software Front Panel Designer
• Cost-effective prototypes and production runs
• Wide range of materials or customization of provided material
• Automatic price calculation
• Fabrication in 1, 3 or 7 days
Sample price: $43.78 plus S&H
FRONT PANEL EXPRESS
www.frontpanelexpress.com
(206) 768-0602

AP CIRCUITS
PCB Fabrication Since 1984
As low as...
$9.95
each!
Two Boards
Two Layers
Two Masks
One Legend
Unmasked boards ship next day!
www.apcircuits.com
VISA
MasterCard
PayPal
IPC MEMBER
ASSOCIATION CONNECTING ELECTRONICS INDUSTRIES
BBB

Lynxmotion
The Lynxmotion Servo Erector Set
Imagine it... Build it... Control it!
With our popular Servo Erector Set you can easily build and control the robot of your dreams. Our interchangeable aluminum brackets make the ultimate precision mechanical assemblies. Our Visual Sequencer for SSC-32 provides powerful PC, Atom, or BS2 based control.
Visit our huge website to see our complete line of aluminum and Lexan based robot kits, electronics, and mechanical components.
Biped BRATs
Round 3DOF Hex
Mini 2DOF Hex
Bipeds
Johnny 5
A4WD1
AL5C Arm
www.lynxmotion.com
Toll Free: 866-512-1024
Images represent a fraction of what can be made!
The SES now has 143 unique components!

AD INDEX

2.4GHz 2 Camera Wireless System

* Channel-scan for multi-camera monitoring
* Night vision (Effective range: 7m)
* Shell, Weather-proof structure for outdoor installation
* Built-in microphones for audio monitoring
* Up to 100m (330ft.) range in open space
* Includes 2 cameras w/power supplies & 1 receiver w/power supply and remote control

Item #
DUAL CAM SYSTEM

$98.50

LED Flashlight with On Board 4GB DVR

Monitoring & Recording concealed in a recharceable flashlight

This unit will function both as a flashlight AND a Digital Video Recorder and has a multitude of uses, It is equiped with a convient USB interface for video data file transfer. You can record color video and then transfer it to a personal computer for viewing. includes two light levels for close up or long distance recording. Also includes a AC adapter for charging the internal Li battery.

Item #
FLASHLIGHT DVR

$189.00

Aardvark

Wireless Inspection Camera

With Color 3.5" LCD Recordable Monitor

Your Extended Eyes & Hands!

See It!
Clearly in narrow spots, even in total darkness or underwater.

Find It!
Fast. No more struggling with a mirror & flash light.

Solve It!
Easily, speed up the solution with extended accessories.

Record It!
With the 3.5" LCD recordable monitor, you can capture pictures or record video for documentation.

Full specifications at www.CircuitSpecialists.com

The Aardvark Wireless Inspection Camera is a video borescope with a 3.5 inch color LCD monitor and a 3ft flexible shaft. The flexible shaft makes the Aardvark great for inspecting hard to reach or confined areas like sink drains, AC Vents, engine compartments or anywhere space is limited. The monitor is wireless and may be separated from the main unit for ease of operation. Still pictures or video can also be recorded and stored on a 2GB MicroSD card (included). The Aardvark comes with attachable mirror, magnet and hook accessories to make seeing or retrieving small items easier. The Aardvark's monitor also has connections for composite video output for a larger monitor/recorder and USB interface for computer connection. Also included is an AC adapter/charger, video cable and USB cable. Optional 3 ft flexible extensions are available to extend the Aardvark's reach (Up to 5 may be added for a total reach of 18 feet!)

Item #
AARDVARK

$249.00

3ft Extension **AARDVARK-EXT** **$24.95**

USB Digital Storage Oscilloscopes

* High performance:
* USB connected: Uses USB and supports plug'n play, with 12Mbp communication speed.
* Best performance for your dollar: Thease units have many features that are comparable to the high speed stand-alone DSOs. But costs a fraction of the price.
* No external power required: Bus-powered from the host computers USB port.
* Probes & USB cable included.
* Easy to use: Intuitive and easy to understand.
* Various data formats: Can save wavrfrom in the following formats: .txt .jpg .bmp & MS excel/word

Specifications	DSO-2090	DSO-2150	DSO-5200
Channels	2 Channels		
Impedence	1M 25pF		
Coupling	AC/DC/GND		
Vertical resolution	8 Bit		9 Bit
Gain Range	10mV-5V, 9 Steps		10mV-10V, 10Steps
DC Accuracy	+/- 3%		
Timebase Range	4ns - 1h 38 Steps		2ns-1h, 39 Steps
Vertical adjustable	Yes		
Input protection	**Diode** clamping		
X-Y	Yes		
Autoset	Yes (30Hz~40MHz)	Yes (30Hz to 60MHz)	Yes (30Hz to 200MHz)
EXT. input	Yes		
Trigger Mode	Auto / Normal / Single		
Trigger Slope	+/-		
Trigger Level Adj.	Yes		
Trigger Type	Rising edge / Falling Edge		
Trigger Source	Ch1 / Ch2 / EXT		
Pre/Post trigger	0-100%		
Buffer size	10K-32K per ch		10K-512KB per ch
Shot Bandwidth	DC to 40MHz	DC to 60MHz	100MHz
Max Sanple Rate	100MS/s	150MS/s	250MS/s
Sampling Selection	Yes		
Waveform Display	port/line, waveform average, persistence, intensity		
Network	open / close		
Vertical Mode	Ch1, Ch2, Dual, Add		
CursorMeasurement	Yes		
Spectrum Analyzer			
Channels	2 Channels		
Math	FFT, addition, subtraction, multiplication, division.		
Bandwidth	40 MHz	60 MHz	200 MHz
Cursor	Frequency, Voltage		
Data Samples	10K-32K/Ch		10K-1M/Ch

40MHz **DSO-2090** **$149.00**

60MHz **DSO-2150** **$169.00**

200MHz **DSO-5200** **$369.00**

60MHz Hand Held Scopemeter with Oscilloscope & DMM Functions

Who Says you can't take it with you?

With the DSO1060 YOU CAN!

You get both a 60 MHz Oscilloscope and a multi function digital multimeter, all in one convenient lightweight rechargeable battery powered package. This power packed package comes complete with scopemeter, test leeds, two scope probes, charger, PC software, USB cable and a convenient nylon carrying case.

- 60MHz Handheld Digital Scopemeter with integrated Digital Multimeter Support
- 60MHz Bandwidth with 2 Channels
- 150MSa/s Real-Time Sampling Rate
- 50Gsa/s Equivalent-Time Sampling Rate
- 6,000-Count DMM resolution with AC/DC at 600V/800V, 10A
- Large 5.7 inch TFT Color LCD Display
- USB Host/Device 2.0 full-speed interface connectivity
- Multi Language Support
- Battery Power Operation

(Installed)

Item #
DSO1060

NEW ITEM

$569.00

Your Power Supply Headquarters!!

We carry a LARGE selection of power supplies from bench top to variacs to single, dual and triple output to wall plug AC adapters to large and ultra large regulated power supplies.

Adjustable DC Power Supplies with Adjustable Current Limiting

NEW ITEM

Regulated linear power supplies with adjustable current limiting. The LED display shows both Volts & Amps. The current output can be preset by the user via a front panel screwdriver adjustment screw while the voltage is adjustable by a front panel multi-turn knob for precise voltage settings. Output is by front panel bananna jacks and there is also a covered terminal strip for remote voltmeter sensing at the load.

* Utilizes SMD technology
* Pre-Settable Voltage & Current levels
* Front Panel On/Off Switch
* Large LED readout for Voltage & Current
* S+ & S- Sampling terminals

0-30 Volt / 0-10 Amp Adj.	(CSI3010X)	$198.00
0-30 Volt / 0-20 Amp Adj.	(CSI3020X)	$299.00
0-40 Volt / 0-10 Amp Adj.	(CSI4010X)	$269.00
0-60 Volt / 0-10 Amp Adj.	(CSI6010X)	$319.00
0-30 Volt / 0-10 Amp Adj.	(CSI3010X)	$198.00
0-120 Volt / 0-3 Amp Adj.	(CSI120030X)	$265.95

www.circuitspecialists.com/dcpower

Energy Saving Variacs

These adjustable Variacs from CircuitSpecialists are VERY popular. They can be used in science labs, by audiophiles (Great for old Tube Amplifiers), hobbiests, in film & video lighting applications. Units are adjustable from 0-130Volts

5Amp	(TDGC2-0.5)	$49.95
20Amp	(TDGC2-2)	$104.00
30Amp	(TDGC2-3KVA)	$119.00

www.circuitspecialists.com/variacs

HengFu Switching Power Supplies

Hengfu

Circuit Specialists carries a wide selection of HengFu switching power supplies. All models have overload, over voltage, over temperature & short circuit protection.

Hi-Power Enclosed Single Output

	Item #	1+	10+
1000W 12v/84A	(HF1000W-SM-12)	$299.00	
1000W 24v/42A	(HF1000W-SM-24)	$299.00	
700W 48V/15A	(HF700W-S-48)	$189.00	$174.00
500W 48V/11A	(HF500W-S-48)	$119.00	
500W 24V/21A	(HF500W-S-24)	$169.00	$149.00
500W 12V/42A	(HF500W-S-12)	$169.00	$149.00
300W 48V/6.3A	(HF300W-S-48)	$ 69.00	$ 61.00
300W 36V/8.5A	(HF300W-S-36)	$ 69.00	$ 61.00
300W 24V/12.5A	(HF300W-S-24)	$ 69.00	$ 61.00
300W 12V/25A	(HF300W-S-12)	$ 69.00	$ 61.00
300W 9V/33A	(HF300W-S-9)	$ 69.00	$ 61.00
300W 5V/60A	(HF300W-S-5)	$ 69.00	$ 61.00

Open Frame Single Output

	Item #	1+	10+
15W 5V/3A	(HF15W-SPL-5)	$ 15.99	$ 13.59
15W 12V/1.3A	(HF15W-SPL-12)	$ 15.99	$ 13.59
15W 24V/0.85A	(HF15W-SPL-24)	$ 15.99	$ 13.59
15W 48V/.032A	(HF15W-SPL-48)	$ 15.99	$ 13.59

Enclosed Dual Output

	Item #	1+	10+
5V/10A-12V/4A	(HF100W-DF-A)	$ 39.00	$ 34.88
5V/15A-15V/2A	(HF100W-DF-B)	$ 39.00	$ 34.88
5V/24A-24V/1A	(HF100W-DF-V)	$ 39.00	$ 34.88
5V/1A-5V1.8A	(HF10W-DL-A)	$ 15.95	$ 13.79
5V/1.8A-12V/0.5A	(HF10W-DL-B)	$ 15.95	$ 13.79
5V/1.8A-15V/0.5A	(HF10W-DL-C)	$ 15.95	$ 13.79
5V/1.8A-24V/0.3A	(HF10W-FL-D)	$ 15.95	$ 13.79

www.circuitspecialists.com/hengfu

POTRANS [Special Purchase]

150Watt 24V/6.5A Switchable Power Supply

* High efficiency
* High reliability
* Protection: Over-voltage/Over-current/Over-power/Short-circuit
* Output reverse protection
* VAC input range selected by switch
* 100% full load burn-in test
* EMI/RFI: FCC Part 15J, Class A & CISPR 22 Class A

Item # CSI-15024-1M
1+$19.00 10+$14.95 100+12.95
UL, cUL, CCC, CE & TUV approved

Programmable DC Power Supplies

- Up to 10 settings stored in memory
- Optional RS-232, USB, RS-485 adapters
- May be used in series or parallel modes with additional supplies.
- Low output ripple & noise
- LCD display with backlight
- High resolution at 1mV

Model	CSI3644A	CSI3645A	CSI3646A
DC Voltage	0-18V	0-36V	0-72V
DC Current	5A	3A	1.5A
Power (max)	90W	108W	108W
Price	$199.00	$199.00	$199.00

Dual Output DC Bench Power Supplies

High stability digital read-out bench power supplies featuring constant voltage & current outputs. Short-circuit & current limiting protection is provided. SMT PC boards and a built-in cooling fan help ensure reliable performance & long life. All 3 Models have a **1A/5VDC Fixed Output** on the rear panel.

Item #:	Price 1+	Price 5+
CSI3003X-5 0-30V/0-3A	$119.00	$112.00
CSI5003X5 0-50V/0-3A	$127.00	$119.00
CSI3005X5 0-30V/0-5A	$129.00	$122.00

Triple Output DC Bench Power Supplies

- Output: 0-30VDC x 2 @ 3 or 5 Amps & 1fixed output @ 5VDC@3A
- Stepped Current: 30mA +/- 1mA

Item #:	Price 1+	Price 5+
CSI3003X3 0-30Vx2@3A	$198.00	$193.00
CSI3005XIII 0-30Vx2@5A	$259.00	$244.00

0-30V / 0-5A . DC Power Supply

NEW ITEM

The CSI530S is a regulated DC power supply which you can adjust the current and the voltage continuously. An LED display is used to show the current and voltage values. The output terminals are safe 4mm banana jacks. This power supply can be used in electronic circuits such as operational amplifiers, digital logic circuits and so on. Users include researchers, technicians, teachers and electronics enthusiasts. A 3 ½ digit LED is used to display the voltage and current values.

ww.circuitspecialists.com/csi530s

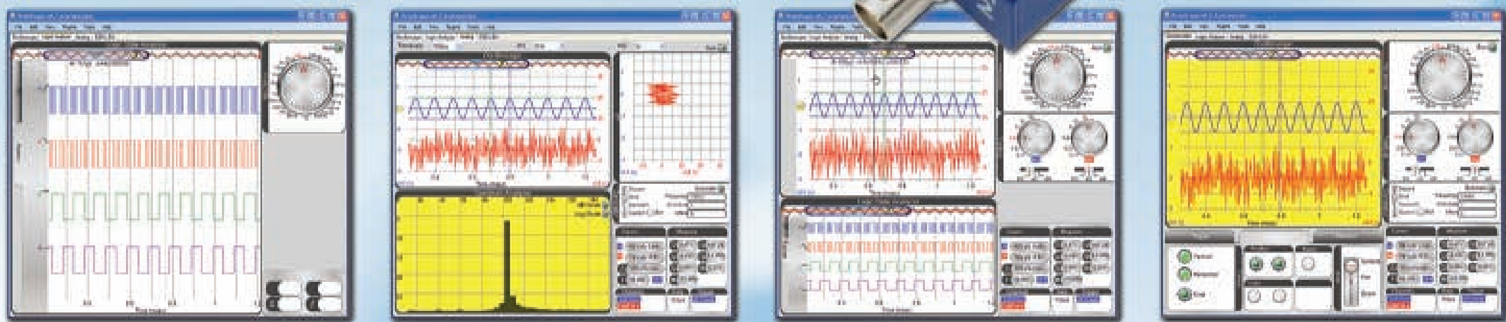
Item # CSI530S $79.00

CircuitSpecialists.com
1000's of Items Online!
1-800-528-1417
Fax: 480-464-5824
Since 1971

PropScope

USB Oscilloscope

The PropScope (#32220; $249.99) is a two-channel oscilloscope that is capable of reading 25 million samples per second with ten bits of resolution over one, two, ten, or twenty volt peak-to-peak waveforms.



Power is provided through the USB port, requiring only a single cable to connect the PropScope to any laptop or desktop PC. The included DAC card provides an analog trigger, a four-bit digital trigger, an eight-bit digital to analog converter, and a four-bit NTSC/PAL output.

The included software provides a traditional scope interface along with auto measurements, the ability to store and export waveforms, a function generator, a logic analyzer, a spectrum analyzer, a vectorscope, and more.

To order the **PropScope USB Oscilloscope (#32220; $249.99)** visit www.parallax.com. Or call our Sales Department toll-free: 888-512-1024 (Monday - Friday, 7 a.m. - 5 p.m., PST).

Prices subject to change without notice. PropScope, Propeller, Parallax and the Parallax logo are trademarks of Parallax Inc.

PARALLAX INC

www.parallax.com

Friendly microcontrollers, legendary resources.